

APLIKASI MOBILE SIPJUHO (SISTEM PRESENSI JURNAL HARIAN ONLINE GURU)

Ulfa Ladayya^{*1)}, Dian Permata Sari²⁾, Suprih Widodo³⁾, Nuur Wachid Abdulmajid⁴⁾, Zia Imam Perdana⁵⁾

1. Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia Kampus di Purwakarta, Indonesia
2. Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia Kampus di Purwakarta, Indonesia
3. Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia Kampus di Purwakarta, Indonesia
4. Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia Kampus di Purwakarta, Indonesia
5. Informatika, Teknologi Informasi dan Komunikasi, Universitas Horizon Indonesia, Indonesia

Article Info

Kata Kunci: Aplikasi *Mobile* SiPJuHOG; Framework Flutter Bahasa Dart; PHP; SQL; *WampServer*

Keywords: *Dart Language Flutter Framework*; SiPJuHOG *Mobile Application*; PHP; SQL; *WampServer*

Article history:

Received 28 Oktober 2024
Revised 18 November 2024
Accepted 1 Maret 2025
Available online 1 Maret 2025

DOI :

<https://doi.org/10.29100/jipi.v10i1.5859>

* Corresponding author.

Ulfa Ladayya
E-mail address:
ladayaulfa@upi.edu

ABSTRAK

Seorang guru mempunyai banyak tanggung jawab, mulai dari kegiatan mengajar hingga memenuhi seluruh administrasi yang ada. Adanya kesenjangan sosial antara ASN dan non-ASN terkait presensi menjadi salah satu permasalahan di Sekolah. Permasalahan lainnya adalah presensi ditulis secara langsung pada agenda kelas, sehingga menyulitkan pihak Sekolah memantau kehadiran guru dan belum terdapat sistem presensi jurnal harian guru di SMKN 2 Purwakarta. Berdasarkan inilah dibangun aplikasi *mobile* SiPJuHOG (Sistem Presensi Jurnal Harian *Online* Guru) untuk menyelesaikan permasalahannya. Tujuan dari penelitian : 1) Mempermudah dan mempercepat guru dalam mengisi presensi jurnal hariannya; dan 2) Mempermudah pihak sekolah merekap kehadiran guru ASN dan non-ASN ketika mengajar. Jenis dan desain penelitian yang digunakan yaitu metode *prototype*. Berdasarkan penelitian diperoleh kesimpulan : 1) Aplikasi *Mobile* SiPJuHOG dibangun menggunakan bahasa dart, *framework* flutter, PHP, SQL, dan menggunakan *wampserver*. Berdasarkan *manual testing* menggunakan “Browserstack” semua fungsi berjalan baik; dan 2) Hasil uji fungsionalitas aplikasi *mobile* kepada *user* berhasil melalui semua instrumen fungsi. Hasil uji *usability* memperoleh nilai total 71,59 dengan kategori “GOOD” (*adjective ratings*), *grade C* (*grade scale*), dan “ACCEPTABLE” (*acceptability ranges*). Sedangkan, uji efisiensi memakai “DevTools” diperoleh *frame per second* rata-ratanya sebesar 58. Artinya aplikasi berjalan cukup baik dalam hal kinerja visualnya. Terakhir ialah uji portabilitas yang dilakukan ketika *user testing mobile* dan menggunakan “BrowserStack” memperoleh hasil sesuai kriteria, yakni berjalan di *handphone* Samsung, Oppo, Vivo, dan Motorola.

ABSTRACT

A teacher has many responsibilities, from teaching activities to fulfilling all existing administration. The existence a social gap between ASN and non-ASN regarding attendance is one of the problems at the School. Another problem is that attendance is written directly on the class agenda, making difficult for the School to monitor teacher attendance and there's no daily teacher journal attendance system at SMKN 2 Purwakarta. Based on this, SiPJuHOG (Teacher Online Daily Journal Presence System) mobile application was built to solve the problem. The purpose of research: 1) Make easier and faster for teachers to fill in their daily journal attendance; and 2) Make easier for schools to recap the attendance of ASN and non-ASN teachers when teaching. The type and design of research used is prototype method. Based on research, conclusions were obtained: 1) SiPJuHOG Mobile Application was built using dart language, flutter framework, PHP, SQL, and using wampserver. Based on manual testing using ‘Browserstack’ all functions run well; and 2) The results of mobile application functionality test to users succeeded through all function instruments. The usability test results obtained total score of 71,59 with the category ‘GOOD’ (*adjective ratings*), *grade C* (*grade scale*), and ‘ACCEPTABLE’ (*acceptability ranges*). Meanwhile, efficiency test using ‘DevTools’ obtained an average frames per second of 58. This means

that the application runs quite well in terms of visual performance. Finally, portability test conducted during mobile user testing and using 'BrowserStack' obtained results according to criteria (running on Samsung, Oppo, Vivo, and Motorola mobile phones).

I. PENDAHULUAN

KEGIATAN belajar mengajar ialah kegiatan rutinitas dari seseorang yang berprofesi sebagai guru di sekolah yang memiliki tanggung jawab yang tidak ringan, dimana selain mengajar, guru juga harus memenuhi seluruh administrasinya. Adanya kesenjangan sosial antara ASN dan non-ASN perihal presensi menjadi salah satu permasalahan yang ada di Sekolah. Permasalahan lainnya adalah masalah presensi guru di Kelas yang ditulis secara langsung pada agenda kelas. Hal ini menjadi permasalahan bagi pihak sekolah, karena pihak sekolah tidak dapat memantau secara langsung kehadiran guru-guru tersebut di kelas.

Keperluan administrasi pendidikan bagi seorang guru tidak dapat dilepaskan dari aktualisasi peran dan fungsi. Salah satu yang harus dipenuhi ialah agenda atau jurnal. Umumnya agenda harian diisi oleh guru secara manual. Kegiatan tersebut mempunyai beberapa dampak yang dapat mempengaruhi kinerja guru. Pengisian jurnal secara manual dinilai kuranglah efektif dan efisien dalam menyokong kemampuan guru dan juga pembuatan laporan agenda harian yang harus diserahkan kepada pihak sekolah [1]. Pekerjaan yang banyak dapat menyebabkan kelelahan, menimbulkan tekanan, dan juga mengganggu waktu untuk merencanakan pembelajaran maupun waktu berinteraksi dengan peserta didik [2]. Banyaknya administrasi yang harus dipenuhi juga dapat memberikan tekanan yang memberikan pengaruh kepada kualitas kinerja guru yang utama yakni mengajar [3].

Agenda atau jurnal harian haruslah dibawa dan diisi oleh guru setiap jam pelajaran di kelas mana pun. Administrasi tersebut adalah salah satu agenda harian yang diperlukan oleh guru dalam aktivitas pembelajaran yang mana dapat membantu dalam manajemen aktivitas pembelajaran guna menggapai tujuan pembelajaran yang diinginkan [4].

Untuk mempermudah dan mempercepat guru dalam mengisi jurnal hariannya, maka akan dibangun aplikasi *mobile* presensi jurnal harian guru. *Mobile* presensi jurnal harian guru yang ingin dibangun terdiri dari dua fitur yakni fitur "Jurnal Harian" dan "Form Pengajuan". Untuk fitur jurnal harian tidak diketik secara luas oleh guru, tetapi di halaman tersebut langsung terdapat fitur kamera untuk bukti presensi jurnal harian yang langsung terdapat keterangan jadwal mengajar (nama, NIP, mata pelajaran), tinggal menambahkan jam awal dan jam akhir, serta terdapat catatan yang dapat diisi oleh guru (opsional). Selain itu, untuk *form* pengajuan sendiri akan memudahkan guru dalam pengajuan cuti dan pengajuan izin lainnya. Di dalam halaman *form user* hanya tinggal memilih jenis izin, memasukkan tanggal awal izin, dan tanggal akhir izin. Selain itu, *user* dapat melihat semua jurnal yang telah diisi di "Jurnal Saya" dan semua *form* yang telah diisi di "Pengajuan Saya". Fitur-fitur tersebut akan ada dibagian sisi *user*. Sedangkan di bagian admin terdapat fitur melihat presensi jurnal harian guru, fitur melihat *form* pengajuan, dan fitur tambah jadwal mengajar.

Berdasarkan wawancara yang telah dilakukan kepada salah satu guru di SMK Negeri 2 Purwakarta bahwasanya saat itu guru masih menggunakan kertas untuk presensi jurnal guru. Padahal seharusnya era sekarang ini kita harus berhenti untuk menggunakan kertas dalam kegiatan sehari-hari di Sekolah (*paperless*). Untuk itu diperlukan sistem yang dapat mempermudah dan mempercepat guru dalam mengisi kehadiran di kelas pada jurnal hariannya serta mempermudah pihak sekolah dalam merekap kehadiran Bapak/Ibu guru ASN dan non-ASN dalam mengajar, karena selama ini khususnya guru non-ASN belum memilikinya. Oleh karena itu, ketika diadakan penilaian guru teladan atau penilaian dalam hal apapun pihak sekolah tidak akan mengalami kesulitan lagi. Dengan semua permasalahan yang ada, maka *quisisioner* I menyetujui dibangunnya sistem presensi jurnal harian *online* guru (SiPJUHOG).

Selanjutnya, berdasarkan wawancara berikutnya yang telah dilakukan kepada *quisisioner* II, bahwasanya presensi jurnal guru masih dilaksanakan secara manual untuk saat ini. Maka dari itu diperlukan sistem yang mempermudah untuk melaksanakan presensi jurnal guru. *Quisisioner* II juga menyetujui dibangunnya sistem tersebut.

Berdasarkan permasalahan yang ada, aplikasi *mobile* ini dibutuhkan untuk membantu guru agar mudah dalam mengisi presensi jurnal hariannya dan memenuhi salah satu administrasinya. Dengan adanya aplikasi ini diharapkan guru dapat mengisi presensi jurnal harian setiap harinya tanpa terlewat. *Mobile* ini juga dibangun untuk mengurangi pemakaian kertas (*paperless*), karena umumnya pengisian jurnal harian menggunakan kertas. *Paperless* ialah salah satu upaya dalam menekan risiko kerusakan pada lingkungan [5]. Dengan berkembangnya teknologi yang pesat ini diharapkan dapat dimanfaatkan untuk memberikan kesadaran kepada semua orang bahwasanya melindungi lingkungan itu penting yakni salah satu caranya dengan mengurangi pemakaian kertas dan menggantinya ke pemakaian teknologi seperti memakai aplikasi *mobile*. Artinya penggunaan *mobile* ini juga mengurangi pemakaian kayu, karena untuk pengolahan kertas tentunya membutuhkan banyak pohon yang harus ditebang

(penebangan pohon akan dilakukan secara terus menerus). Pengurangan pemakaian kertas juga sudah dilakukan di beberapa sekolah. Contohnya di dalam jurnal milik Maesaroh, Bahagia, dan Kamalludin [6] dijelaskan bahwasanya MAN 2 Kota Bogor melakukan kerja sama dengan sekolah-sekolah swasta untuk melakukan pembinaan dan juga pelatihan dalam mengelola dan melestarikan lingkungannya yang mana usaha yang dilakukan sekolah tersebut ialah mengurangi penggunaan kertas untuk semua tugas Sekolah. Selain itu, di dalam jurnal milik Megasyah dan Hardian [7] juga dijelaskan pengurangan pemakaian kertas yang mana sesudah dibuatnya sistem yang baru, para pustakawan tidak memakai kertas dalam mencatat dan juga menyerahkan data sirkulasi serta transaksi kepada kepala sekolah sehingga mengurangi penumpukan berkas-berkas. Tidak hanya menurunkan pemakaian kertas, penggunaan *mobile* ini juga dapat mengurangi sampah. Itu disebabkan karena umumnya penggunaan kertas akan menyumbangkan banyaknya sampah pada lingkungan sekitar. Terakhir, pemakaian *mobile* juga dapat menghemat energi, karena tidak menggunakan perangkat keras yang dapat memicu pengurangan energi ataupun emisi karbon, karena teknologi informasi *mobile* menggunakan penyimpanan seperti komputasi awan.

Pembuatan sistem jurnal harian guru sudah diteliti oleh beberapa peneliti sebelumnya dengan permasalahan yang berbeda-beda begitu juga dengan permasalahan yang dimiliki peneliti itu berbeda juga dengan peneliti lainnya. Menurut Muslimin dan Sopian aplikasi jurnal dibuat untuk memasukkan agenda harian dan menyediakan aplikasi untuk manajemen agenda harian yang telah diinputkan. Fitur-fitur yang terdapat di *user* ialah *form login*, *form* pendaftaran, *form* utama, *form* agenda, *form* lihat jadwal, *form* lihat agenda, *form* tentang, *form* logout. Sedangkan aplikasi admin memiliki fitur *form login*, *form* menu utama, *form* data *user*, *form* tambah *user*, *form* jadwal, *form* tambah jadwal, dan *form* agenda. Selain itu, terdapat pula aplikasi lainnya yang berbasis android milik Nasrulloh dan Agustina [8] dijelaskan bahwasanya aplikasi CPPH dibuat agar guru dapat mencatat pembelajaran harian dan mempermudah dalam memberikan informasi ketika melakukan pembelajaran agar lebih efisien. Fitur-fitur yang ada di *website* tersebut adalah *login*, halaman utama admin atau kepala sekolah, halaman utama guru. Selanjutnya, menurut Anjeli dan Yanto [9] agenda guru sangatlah esensial yang mana berguna untuk menunjang tugas-tugas guru agar lebih mudah untuk melaksanakan tugas seorang guru dalam mencatat agendanya. Fitur-fitur yang tersedia di dalam web jurnal mengajar tersebut ialah *login* guru dan *admin*, tampilan agenda kegiatan, tampilan *search*, *refresh*, dan *print* agenda kegiatan. Selain itu, Junaidi dan Ghofur [10] menjelaskan terdapat sekolah yang melaksanakan beberapa proses dengan cara yang manual, yakni salah satunya pengolahan dalam penulisan jurnal pengajaran (proses manual tersebut salah satunya dilaksanakan di SMKN 1 Sungai Pinang). Dimana mekanisme persiapan jurnal mengajar yang akan diberikan kepada peserta didik belum dilakukan persiapannya dan menjadi sedikit terhambat penyampaiannya. Lalu, sering terjadi permasalahan dalam aktivitas operasionalnya. Maka untuk menyelesaikan semua permasalahan tersebut, beliau membuat aplikasi web dengan fitur *login*, data guru, data mata pelajaran, data kelas, data agenda, data *user*, dan laporan. Terakhir di dalam jurnal milik Setiawan, Waluyo, dan Suseno [11] dijelaskan bahwasanya dampak dari covid 19 sangat berpengaruh ke dalam berbagai bidang yang salah satunya bidang Pendidikan dengan pembelajaran daring yang ditetapkan oleh peraturan Pendidikan. Dengan begitu, guru tetap diminta untuk membuat administrasi agenda hariannya saat pembelajaran di rumah yang saat ini masih berupa buku dan permasalahan ini menyulitkan untuk pengecekan dan pengawasan oleh kepala sekolah. Berdasarkan semua permasalahan tersebut, maka beliau membuat perancangan sistem informasi agenda kegiatan guru berbasis *website*. Sistem tersebut mempunyai fitur *login*, tampilan *dashboard* guru mapel, tampilan agenda pengajaran guru mapel, dan tampilan *view* agenda guru mapel.

Berdasarkan dari artikel jurnal yang telah dibuat oleh beberapa peneliti sebelumnya terlihat beberapa perbedaan antara fitur-fitur yang telah dibuat oleh peneliti lainnya dengan fitur yang akan dibuat oleh peneliti sekarang. Perbedaannya adalah peneliti sekarang akan membuat fitur kamera untuk bukti presensi jurnal harian yang langsung terdapat keterangan jadwal mengajar (nama, NIP, mata pelajaran), tinggal menambahkan jam awal dan jam akhir, serta terdapat catatan yang dapat diisi oleh guru (opsional). Selain itu, untuk *form* pengajuan sendiri akan memudahkan guru dalam pengajuan cuti dan pengajuan izin lainnya. Di dalam halaman *form user* hanya tinggal memilih jenis izin, memasukkan tanggal awal izin, dan tanggal akhir izin. Selain itu, *user* dapat melihat semua jurnal yang telah diisi di “Jurnal Saya” dan semua *form* yang telah diisi di “Pengajuan Saya”. Fitur-fitur tersebut akan ada dibagian sisi *user*. Sedangkan di bagian admin terdapat fitur melihat presensi jurnal harian guru, fitur melihat *form* pengajuan, dan fitur tambah jadwal mengajar.

Pada dasarnya, digitalisasi untuk administrasi guru di Sekolah bukanlah hal yang baru di dunia teknologi informasi, karena jauh sebelumnya penyelesain administrasi secara digital dengan teknologi informasi sudah dimiliki oleh beberapa instansi atau perusahaan dibidang yang berbeda-beda. Seperti program pemerintah yang telah dikembangkan yakni e-Government [12] yang mana secara konsepnya ialah memberikan pelayanan secara elektronik atau e-service. Selain itu, terdapat contoh lainnya yakni SIMRS (Sistem Informasi Manajemen Rumah Sakit) [13] yang mana sistem tersebut sudah melakukan pengintegrasian bisnis rumah sakit dengan pelayanan yang di dalamnya termasuk semua mekanisme administrasi bagian *front office* dan *back office*, mekanisme inventaris, penggunaan dan pembelian barang, serta kegiatan untuk memberikan *report* melalui *dashboard business*

intelligence dari sistem tersebut. Terdapat juga sistem lainnya yang mengatasi masalah administrasi. Sistem tersebut di buat untuk sebuah perusahaan [14], yakni sistem *mobile* yang memudahkan perusahaan dalam kegiatan administrasi dan pelanggan atau pengecer dalam melaksanakan transaksi pembelian pupuk, kemudian aktivitas pendistribusian dapat ditinjau dn juga mekanisme dalam pemberitaan administrasi dapat menjadi lebih mudah dan cepat (menekan semua kendala yang terjadi).

II. METODE PENELITIAN

Dalam membuat artikel ini, jenis dan desain penelitian yang dipakai ialah metode *prototype*. Metode *prototype* adalah paradigma baru dalam pengembangan sistem maupun aplikasi yang mana tidak hanya selaras dengan sebuah evolusi dari metode pengembangan sistem informasi, tetapi juga merupakan revolusi dalam sebuah pengembangan sistem informasi itu sendiri [15]. Metode *prototype* merupakan metode yang banyak dipakai oleh *developer* agar dapat saling terkait dengan *user* selama proses pembuatan sistem nya [16]. Lalu, instrumen penelitian menggunakan *instrument* non tes yakni wawancara. Sedangkan teknik yang digunakan untuk menguji aplikasi *mobile* ini adalah teknik *black box testing*. *Black box testing* yang dimaksud adalah pengujian yang berlandaskan kepada detail-detail dari aplikasi contohnya seperti tampilan aplikasi, fungsi-fungsi dari aplikasi, dan juga kesesuaian alur fungsi dengan sistem yang diinginkan oleh *developer* nya. Teknik *black box testing* yang digunakan adalah fungsional *testing* dengan menggunakan *website* “BrowserStack” yang memiliki fitur “App Live” (*manual testing*). Pemilihan *website* BrowserStack untuk pengujian manual dilakukan karena web ini mempunyai keunggulan yakni mempunyai fitur “App Live” yang mana saat pengujian dilakukan serasa pengujian di *handphone* secara langsung. Yang membuatnya unggul untuk proyek ini adalah karena dapat memasukkan *mobile* yang sudah *release* untuk diuji manual. Setelah di *upload* di dalam halaman *handphone* yang dipilih, maka akan otomatis terunduh di *handphone* nya dan langsung dapat diuji. Skenario pengujian yang dilakukan hanya mengiktui alur penggunaan aplikasi *mobile* SiPJUHOG dari awal hingga akhir (baik *user* ataupun *admin*). Sedangkan kriteria keberhasilan ditentukan dari berhasil tidaknya menyelesaikan skenario tersebut. Teknik *black box testing* dipilih untuk dipakai dalam pengujian aplikasi *mobile* SiPJUHOG disebabkan oleh *timeline* pengerjaan proyek yang singkat. Oleh karena waktu yang dimiliki sangat sedikit, maka pemilihan jenis teknik jatuh kepada *black box testing* , karena teknis ini mempunyai jenis fungsional *testing*. Seperti yang diketahui fungsional *testing* hanya menguji fungsiona dari aplikasi *mobile* saja yang mana memungkinkan untuk dipakai karena tidak membutuhkan waktu yang lama dalam pengujiannya. Keunggulan teknik tersebut dalam pengujian aplikasi *mobile* adalah dapat mengetahui *mobile* yang telah dibangun sudah dapat berjalan dengan baik atau tidak.

Untuk analisis kualitas yang digunakan adalah model ISO/IEC 9126. Penggunaan model ISO/IEC 9126 untuk menganalisis kualitas disebabkan oleh kemampuan model ini untuk merepresentasikan kualitas perangkat lunak, model, karakteristik dari mutu, dan metrik yang terkait untuk menilai dan memutuskan kualitas sebuah produk *software*. Dimensi-dimensi kualitas yang dinilai hanya empat, yakni fungsionalitas, kegunaan (*usability*), efisiensi, dan portabilitas. Untuk uji yang pertama (fungsionalitas) diukur dari keberhasilan setiap fitur yang dicoba *user* yang akhirnya dihitung menggunakan persamaan (1), kemudian pengujian *usability* diukur dengan memberikan item pertanyaan SUS. Lalu, pengujian efisiensi diukur dari performance *mobile* menggunakan Dev-Tools. Sedangkan untuk pengujian portabilitas diukur dari aplikasi *mobile* SiPJUHOG yang telah dibuat apakah dapat berjalan di beberapa jenis *handphone* yang ditentukan. Adapun tahapan metode *prototype* adalah sebagai berikut.

A. Communication

Tahap pertama dari metode *prototype* ini adalah komunikasi (*communication*). Komunikasi adalah lengkap yang digunaan *developer* atau pengembang bersama klien melaksanakan pembicaraan untuk memperoleh hasil yang diperlukan oleh klien.

B. Quick Plan

Tahap keduanya adalah *quick plan*. Pada tahap ini dilakukan analisis terhadap kebutuhan pengguna. Di dalam tahap ini juga dilakukan perencanaan dan pemodelan dengan cepat yang difokuskan pada semua aspek yang akan dilihat oleh *user*.

C. Modelling Quick Design

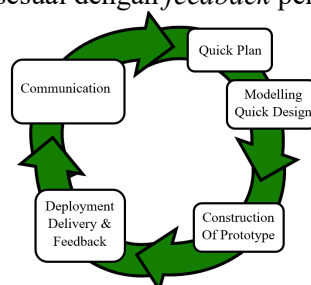
Tahap ketiga dalam metode prototipe adalah *modelling quick design*. Di dalam tahap ini dilaksanakan pembuatan desain yang mana akan memberikan gambaran singkat dari sistem yang akan dibangun. Contohnya seperti pembuatan proses bisnis sistem.

D. Construction Of Prototype

Pada tahap keempat ini dilakukan pembuatan kerangka desain (*wireframe*), *design user interface* (UI), dan juga menafsirkan hasil perancangannya ke dalam bahasa pemrograman sehingga dapat dimengerti oleh komputer.

E. *Deployment Delivery & Feedback*

Pada tahap kelima ini dilakukan *deployment* sistem, pengujian sistem yang telah dibangun, umpan balik dari *user*, dan analisis kualitas sistem. Pengujian dilakukan untuk mengetahui apakah sistem berjalan sesuai dengan keperluan dan tujuan yang diinginkan. Di dalam tahap *deployment*, mekanisme *manual testing* menggunakan BrowserStack (fitur App Live) akan dilaksanakan. Sedangkan jika ada *feedback* dari pengguna setelah pengguna mencoba *mobile* SiPJuHOG, maka umpan balik akan diterima dan segera di proses untuk diperbaiki langsung di dalam aplikasi *mobile* SiPJuHOG nya (ini merupakan perencanaan tindak lanjut berdasar kepada *feedback* dari pengguna untuk disesuaikan lagi dengan sistem *mobile* nya). Artinya ada proses iterasi di dalam pembuatan aplikasi *mobile* ini, yang mana setelah didapatkan umpan balik, maka akan diperbaiki *source code* dari proyek *mobile*, agar sesuai dengan *feedback* pengguna.



Gambar. 1 Metode Prototype

III. HASIL DAN PEMBAHASAN

A. *Communication*

Pada tahap pertama metode *prototype* dilakukan *communication*. Dalam hal ini komunikasi dilakukan dengan melaksanakan wawancara kepada dua narasumber di SMKN 2 Purwakarta. Setelah wawancara dilakukan dapat dilakukan analisis permasalahan yang dialami oleh SMK Negeri 2 Purwakarta, yakni adanya kesenjangan sosial antara ASN dan non-ASN perihal presensi menjadi salah satu permasalahan yang ada di Sekolah. Permasalahan lainnya adalah masalah presensi guru di Kelas yang ditulis secara langsung pada agenda kelas. Hal ini menjadi permasalahan bagi pihak sekolah, karena pihak sekolah tidak dapat memantau secara langsung kehadiran guru-guru tersebut di kelas. Selain itu, di Sekolah tersebut juga belum terdapat sistem presensi jurnal harian guru yang mana presensi guru di kelas belum dapat terpantau langsung oleh pihak sekolah.

B. *Quick Plan*

Pada tahap ini dilakukan analisis terhadap sistem yang mana akan dilakukan peninjauan proses dari sistem yang dimiliki dan menganalisa mekanismenya, serta menganalisis keperluan sistem yang akan dibangun.

1. Memahami dan Menganalisa

Pada langkah ini haruslah dipahami dan dianalisa sistem yang ada di SMK Negeri 2 Purwakarta. Perlu diketahui bahwasanya Sekolah belum mempunyai sistem untuk presensi jurnal harian guru. Dari awal Sekolah masih mempunyai sistem manual untuk mengelola presensi jurnal harian guru. Seperti hasil wawancara yang telah dilakukan kepada Bapak Zia Imam Perdana, S.Kom., M.Kom. bahwasanya sekarang ini guru-guru masih memakai kertas untuk presensi jurnal gurunya (dilakukan secara manual). Padahal seharusnya era sekarang ini kita harus berhenti untuk menggunakan kertas dalam kegiatan sehari-hari di Sekolah (*paperless*). Beliau juga menjelaskan bahwasanya presensi jurnal harian secara manual menggunakan kertas itu menyusahkan. Kesulitannya ialah ketika menjelaskan dibagian pembahasan. Menurut beliau alangkah lebih baiknya jika langsung ada pilihan. Pembuatan jurnal secara manual ini juga menyebabkan beliau sering lupa untuk menulis jurnal harian. Sejalan dengan hasil wawancara pertama, wawancara kedua dilakukan kepada Bapak Cep Kusaeri, S.T. Beliau juga menjelaskan bahwasanya presensi jurnal guru masih dilaksanakan secara manual untuk saat ini. Menurut beliau pengisian jurnal secara manual mempunyai kelebihan dan kekurangan. Beliau mengatakan jarang sekali lupa untuk mengisi presensi jurnal guru, tetapi sekali-kali beliau lupa juga dalam mengisi jurnal tersebut. Berikut sistem manual yang berjalan di Sekolah.

a) Prosedur Sistem Manual Jurnal Harian

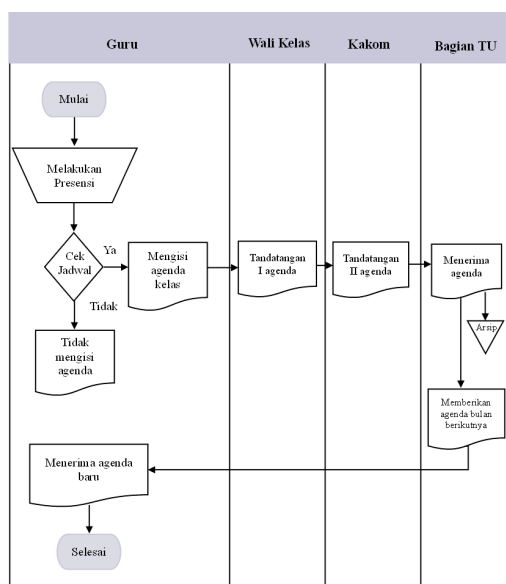
Berikut merupakan prosedur dari pengisian agenda guru secara manual :

- 1) Guru melakukan presensi di kantor guru, kemudian melihat jadwal mengajarnya dimana.

- 2) Di kelas guru mengisi agenda kelas, yang diisi oleh guru seperti, nama guru, nama mata pelajaran, uraian materi, dan tanda tangan guru tersebut.
- 3) Lalu, sekitar seminggu sekali wali kelas akan menandatangani agenda kelasnya.
- 4) Setelah itu, Kepala Program Keahlian menandatangani agenda kelasnya setiap satu bulan sekali.
- 5) Selanjutnya, setiap satu bulan sekali agenda kelas akan di serahkan ke Bagian Tata Usaha dan diberikan agenda baru untuk bulan berikutnya.

b) *Flowmap* Sistem Manual Jurnal Harian

Dari gambar *flowmap* yang ada di gambar 1 dijelaskan bahwasanya alur sistem manualnya dimulai dari guru, wali kelas, kakom, dan bagian TU. Tanda mulainya diawali dari bagian guru, kemudian aksi yang dilakukan ialah melakukan presensi, kemudian mengecek jadwal. Di sini bagan alur keputusan melakukan tugasnya, yakni jika saat cek jadwal ada jam mengajar, maka harus mengisi agenda kelas, dan jika tidak ada tidak mengisi agenda kelas. Saat guru memiliki jam mengajar, maka alur nya maju ke pengisian agenda kelas, kemudian mendapatkan tanda tangan pertama dari wali kelas dan tanda tangan kedua dari kakom (tanda tangan dari wali kelas dan kakom menyesuaikan waktu kapan dapat menandatangani). Lalu, jika sudah sebulan penuh, agenda nya akan diserahkan ke bagian TU (diarsipkan) dan diberikan agenda baru kepada guru. Berikut merupakan *flowmap* dari pengisian agenda guru secara manual.



Gambar. 2 *Flowmap* Sistem Manual Jurnal Harian

2. Analisis Keperluan Sistem

Langkah selanjutnya ini adalah menganalisa kebutuhan dari sistem yang akan dibangun. Saat wawancara dilakukan kepada Bapak Zia Imam Perdana, S.Kom., M.Kom. dan Bapak Cep Kusaeri, S.T., peneliti menanyakan perihal bagaimana jika dibangun aplikasi *mobile* untuk mengatasi semua permasalahan tersebut. Ide untuk membuat aplikasi *mobile* SiPJUHOG ini mendapatkan respon yang sangat baik dari beliau-beliau. Bapak Zia Imam Perdana, S.Kom., M.Kom. secara langsung menyetujui untuk dibangunnya *mobile* ini. Untuk warna *mobilenya*, beliau menyarankan dengan menggunakan warna hijau dan kuning sesuai dengan logo SMK Negeri 2 Purwakarta. Selaras dengan Narasumber yang pertama, Bapak Cep Kusaeri, S.T. juga menanggapi ide ini dengan sangat senang dan setuju. Beliau menyarankan warna aplikasinya hijau dan kuning sesuai logo sekolah, tetapi jika warnanya menjadi biru juga tidak menjadi sebuah permasalahan.

Persetujuan pembuatan *mobile* SiPJUHOG ini tentunya karena beliau-beliau merasa sistem ini dapat sangat membantu guru juga pihak sekolah mengenai presensi jurnal harian guru. Mengenai fitur-fitur untuk *user*, peneliti mengajukan sistem yang akan dibuat ini mengandung fitur *sign up user*, *login user*, *profile user*, *logout user*, dan dua fitur berikut :

a. Fitur Jurnal Harian

- a. Sistem *mobile* SiPJUHOG menyediakan *button* “Isi Jurnal” dan *button* “Jurnal”. *Button* isi jurnal untuk mengisi presensi jurnal harian guru. Sementara *button* jurnal saya berisi hasil jurnal yang telah diisi oleh *user* (guru).
- b. Sistem *mobile* SiPJUHOG menyediakan pemilihan kelas dan hari yang mana ketika di klik akan muncul item lainnya (penambahan foto dari *camera*, tanggal yang akan langsung muncul, memilih jam awal

dan jam akhir mengajar, melihat mata pelajaran yang sudah muncul, dan dapat memberikan catatan (ini bersifat opsional)).

- c. Sistem *mobile* SiPJUHOG memberikan fitur pilih foto dari *camera* yang berguna agar guru dapat mengambil foto secara langsung sebagai bukti presensi jurnal hariannya saat melakukan KBM di Kelas.
 - d. Sistem *mobile* SiPJUHOG memberikan fitur *preview* yang akan muncul setelah *user* klik “Selanjutnya”. Isi *preview* nya adalah jurnal yang telah diisi oleh pengguna.
 - e. Sistem *mobile* SiPJUHOG menyediakan *button* “Kembali” digunakan ketika terdapat kekurangan atau kesalahan (balik ke halaman sebelumnya) dan *button* “Kirim” untuk mengirim jurnal hariannya.
- b. Fitur *Form* Pengajuan
- 1) Sistem *mobile* SiPJUHOG menyediakan *button* “Ajukan” dan *button* “Form Saya”. *Button* ajukan untuk mengisi *form* pengajuan guru. Sementara *button form* saya berisi hasil *form* yang telah diisi oleh *user* (guru).
 - 2) Sistem *mobile* SiPJUHOG menyediakan *form* yang akan muncul otomatis nama, nip, dan jabatannya, lalu memilih jenis izin, tanggal mulai izin, dan tanggal akhir izin.
 - 3) Sistem *mobile* SiPJUHOG memberikan fitur *preview* yang akan muncul setelah *user* klik “Selanjutnya”. Isi *preview* nya adalah *form* pengajuan yang telah diisi oleh *user*.
 - 4) Sistem *mobile* SiPJUHOG menyediakan *button* “Kembali” yang digunakan ketika terdapat kekurangan atau kesalahan (balik ke halaman sebelumnya) dan *button* “Kirim” untuk mengirim pengajuan

Selanjutnya mengenai fitur-fitur untuk *admin*, peneliti mengajukan sistem yang akan dibuat ini mengandung fitur *sign up user*, *login user*, *profile user*, *logout user*, dan tiga fitur berikut.

a. Fitur Lihat Jurnal Harian

Di dalam fitur ini *admin* hanya melihat semua jurnal harian yang telah diisi oleh semua guru berdasarkan hari dan kelas yang dipilih.

b. Fitur Lihat Pengajuan

Selanjutnya *admin* hanya dapat melihat semua pengajuan yang telah diisi oleh guru-guru dan terdapat juga fitur pencarian nama guru.

c. Fitur Penambahan Jadwal Mengajar

Fitur terakhir ialah menambahkan jadwal mengajar yang mana *admin* dapat menambahkan sesuai jadwal guru di semua jurusan yang ada.

C. Modelling Quick Design

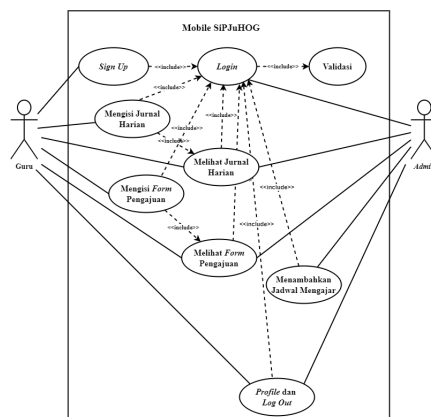
Dalam tahap ini dilakukan pembuatan desain proses bisnis sistem menggunakan UML (*Unified Modelling Language*) yang terdiri dari *use case diagram* dan *class diagram*. Pembuatan jenis-jenis UML tersebut dilaksanakan di dalam *software draw.io*.

1. Proses Bisnis Sistem

Proses bisnis sistem yang dipakai untuk menggambarkan aplikasi *mobile* SiPJUHOG ini adalah *Unified Modelling Language* (UML). Berikut semua merupakan jenis-jenis dari UML yang dipakai.

a. *Usecase Diagram*

Dapat dilihat dari gambar 2 berikut bahwasanya guru (*user*) dapat mengakses *sign up*, *login*, fitur pengisian jurnal harian & melihat jurnal harian, dan fitur pengisian *form* pengajuan & melihat *form* pengajuan. Sedangkan *admin* dapat mengakses *login*, melihat jurnal harian, melihat *form* pengajuan, menambahkan jadwal mengajar. Selain itu, guru dan *admin* juga dapat mengakses *profile* dan *log out* nya di bagiannya masing-masing.

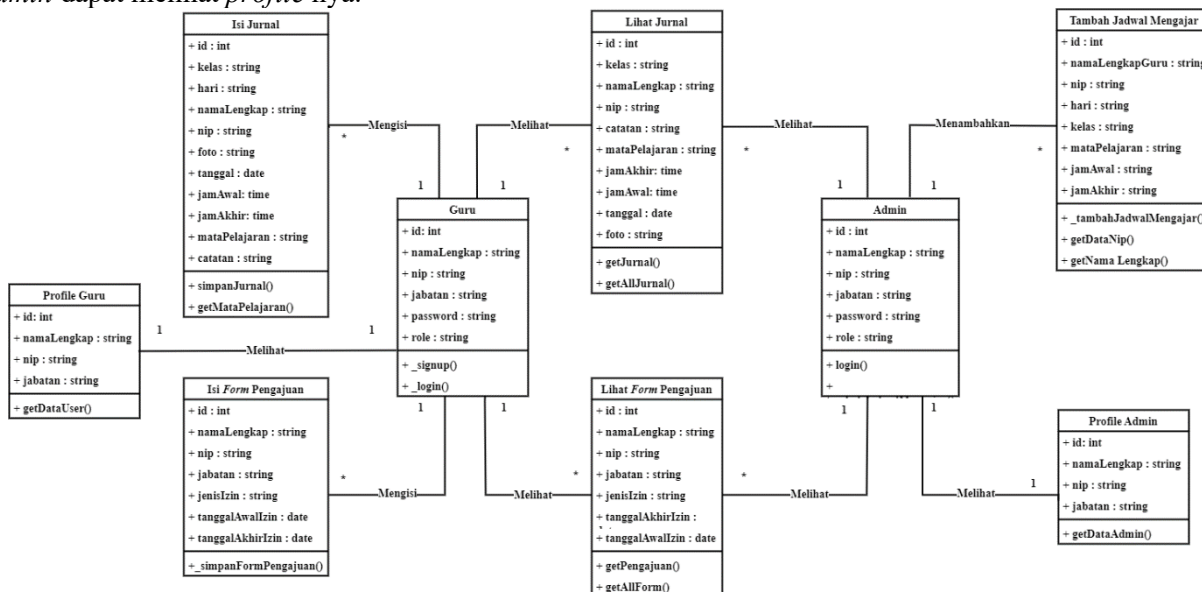


Gambar. 3 *Usecase Diagram* Mobile SiPJUHOG (User dan Admin)

b. Class Diagram

Class diagram digunakan untuk merancang database yang akan digunakan saat pembuatan mobile. Dapat dilihat dari gambar 3 bahwasanya guru (user) dapat sign up dan login untuk masuk ke aplikasi mobile. Selain itu, user juga dapat mengisi banyak jurnal dan banyak jurnal dapat diisi oleh satu guru, begitu juga dengan pengisian form pengajuan. Selanjutnya, satu user dapat melihat banyak jurnal harian yang telah diisinya dan banyaknya jurnal yang sudah diisi dapat dilihat oleh satu user saja, begitu juga saat melihat form pengajuan. Terakhir, user dapat melihat profile nya.

Sedangkan admin hanya melakukan login untuk masuk ke mobile yang seterusnya satu admin dapat melihat semua jurnal harian yang telah diisi oleh semua guru (user) dan semua jurnal tersebut dapat dilihat oleh satu admin, begitu juga dengan melihat form pengajuan. Fitur berikut nya hanya ada di admin, yakni satu admin dapat menambahkan banyak jurnal, tetapi banyak jurnal hanya diisi oleh satu admin. Terakhir, admin dapat melihat profile nya.



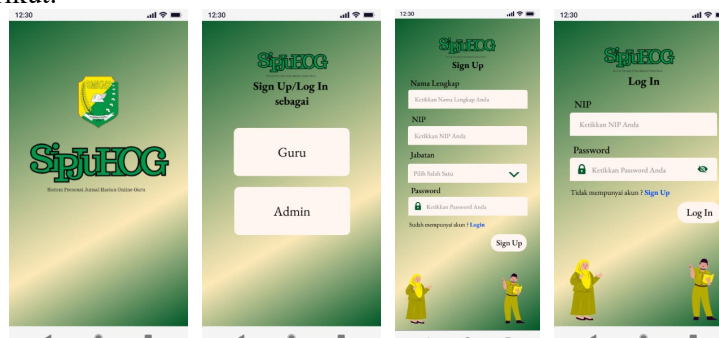
Gambar. 4 Class Diagram Mobile SiPJuHOG (User dan Admin)

D. Construction Of Prototype

Pada tahap ini dilakukan pembuatan kerangka desain (wireframe), design user interface (UI), dan juga menafsirkan hasil perancangannya ke dalam bahasa pemrograman atau membuat pengkodean sehingga dapat dimengerti oleh komputer. Untuk pembuatan wireframe dan user interface dilakukan di figma.

1. Design User Interface (UI) Sistem Aplikasi Mobile SiPJuHOG

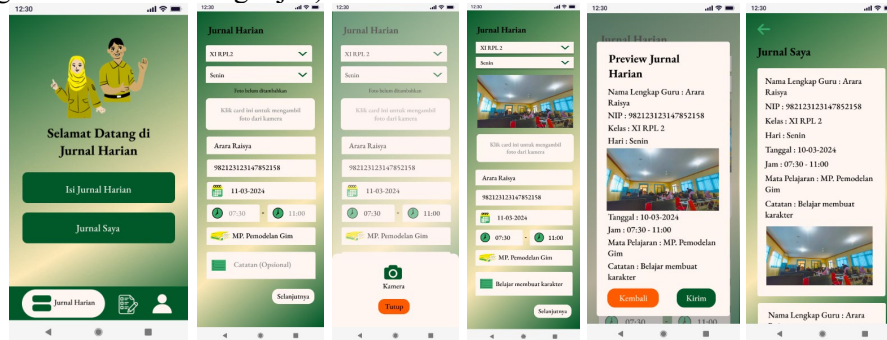
Untuk desain user nya sendiri peneliti membagi desain nya ke dalam dua bagian sesuai role nya, sebagai user (guru) dan sebagai admin. Setelah membuat wireframennya, langkah selanjutnya ialah membuat desain user interface berdasarkan kerangka yang sudah dibangun sebelumnya. Sama seperti tempat membuat wireframe sebelumnya, untuk pembuatan UI juga dilaksanakan di dalam software figma. Perlu diketahui bahwasanya Figma merupakan alat (tool) yang umumnya dipakai oleh seseorang yang membuat design tampilan mobile, website, desktop, dan lainnya [17]. Sedangkan untuk design UI user (guru) yang telah dibangun dapat dilihat dari gambar-gambar berikut.



Gambar. 5 Desain UI User Splash Screen, Pembagian Role, Halaman Sign Up, dan Halaman Login

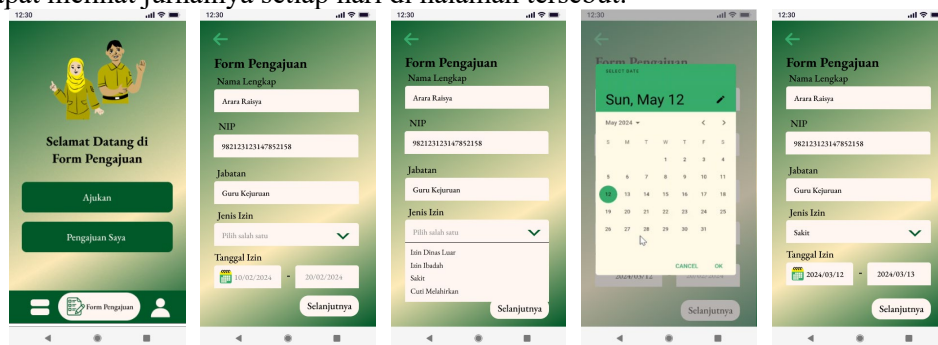
Ketika user pertama kali membuka aplikasi mobile, user akan melihat splash screen yang ditandai dengan logo SMK Negeri 2 Purwakarta dan di bawah nya terdapat logo dari mobile yakni logo “SiPJuHOG”. Lalu,

halaman selanjutnya ada pembagian *role* yang mana jika yang sedang membuka *user* pilih *card* “Guru” (selanjutnya akan *signup* terlebih dahulu dan barulah *login*). Selanjutnya setelah *user* memilih *card* “Guru”, *user* akan berada di halaman *sign up*. Di sini ada empat hal yang harus diisi oleh *user*, yakni nama lengkap, NIP, jabatan, dan *password*. Setelah mengisi semuanya, klik saja *button sign up*. Lalu, *user* akan melihat pemberitahuan bahwasanya berhasil *sign up* yang mempunyai latar belakang berwarna hijau. Setelah *user* berhasil *signup*, *user* hanya mengisi NIP dan *password* yang sudah didaftarkan sebelumnya. Terakhir, *user* klik *button “Log In”* dan akan diarahkan ke halaman inti dari *mobile* ini (akan ada pemberitahuan bahwasanya berhasil *login* dengan latar belakang hijau).



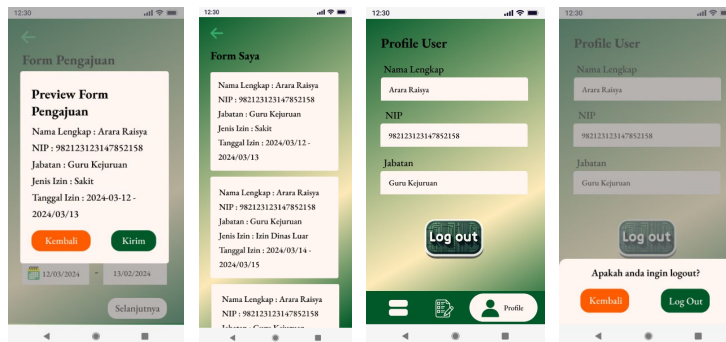
Gambar. 6 Desain UI *User* Fitur Jurnal Harian dan Fitur Jurnal Saya

Fitur ini akan dilihat *user* setelah berhasil *signup* dan *login*. Klik saja *card* “Isi Jurnal Harian” untuk mengisi jurnalnya. Setelah masuk ke halaman jurnal, maka *user* harus memilih kelas dan hari dari *dropdown* yang disediakan. Setelah *user* memilih kelas dan hari, maka otomatis muncul *form* lainnya. Selanjutnya *user* dapat melihat *card* untuk mengambil foto dari kamera, muncul nama & NIP, muncul tanggal sesuai harinya, dan muncul juga mata pelajarannya. *User* dapat mengambil foto langsung dari kamera dengan klik *card* yang terdapat tulisan “Klik *card* ini untuk mengambil foto dari kamera”, kemudian muncul *icon* kamera seperti di gambar. Selanjutnya klik *icon* tersebut, maka akan berlanjut ke kamera hp masing-masing *user* dan tampil di halaman jurnal. Jika ingin menggantinya, klik saja *card* yang terdapat tulisan seperti yang telah dijelaskan sebelumnya. Terakhir, *user* hanya tinggal memilih jam awal, jam akhir, dan mengisi catatan (opsional). Setelah selesai mengisi jurnal, klik *button* “Selanjutnya”, kemudian akan melihat *preview* jurnal harian. Tekan *button* “Kembali” jika terdapat kesalahan dalam pengisian jurnal dan jika sudah benar klik “Kirim”, maka selanjutnya akan ada pemberitahuan yang menyatakan bahwasanya jurnal telah ditambahkan yang mempunyai latar belakang berwarna hijau. *User* dapat melihat semua jurnal yang telah diisi dengan cara klik *card* “Jurnal Saya”, maka *user* dapat melihat jurnalnya setiap hari di halaman tersebut.



Gambar. 7 Desain UI *User* Fitur Form Pengajuan

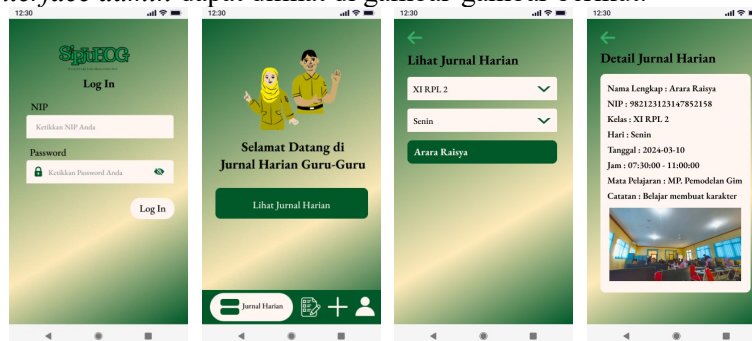
Fitur *user* yang selanjutnya ialah “Form Pengajuan”. Di dalam fitur ini, *user* dapat melihat dua *card*, yakni *card* “Ajukan” dan *card* “Pengajuan Saya”. Untuk mengisi *form* pengajuan, klik *card* “Ajukan” dan akan melihat halaman *form* pengajuan. Di halaman ini nama lengkap, NIP, dan jabatan akan langsung muncul. Artinya *user* hanya akan memilih jenis izin, memilih tanggal mulai izin, dan memilih tanggal akhir izin saja. Pemilihan jenis izin nya terbagi menjadi 4 bagian, yakni izin dinas luar, izin ibadah, sakit, dan cuti melahirkan. *User* dapat memilih jenis izin nya sesuai dengan kebutuhan *user* saat itu. *User* dapat memilih tanggal izin nya dimulai dari tanggal berapa hingga tanggal berapa dari dua *form* yang sudah disediakan.



Gambar. 8 Desain UI *User Preview Form Pengajuan*, *Fitur Pengajuan Saya*, dan *Fitur Profile*

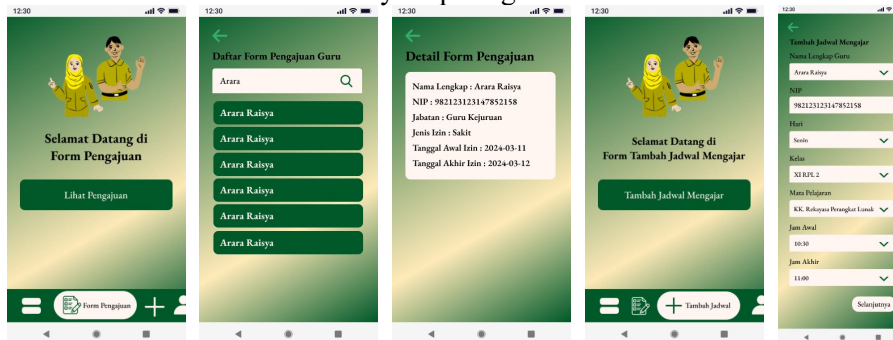
Setelah *user* selesai mengisi *form* nya, maka *user* dapat menekan *button* “Selanjutnya” dan akan melihat *preview*. Jika ada kesalahan dalam pengisiannya dapat menekan *button* “Kembali” dan jika sudah benar dapat menekan *button* “Kirim”. Selanjutnya *user* akan melihat pemberitahuan bahwasanya *form* pengajuan berhasil ditambahkan. Untuk melihat semua *form* yang sudah pernah diajukan, pilihlah *card* “Pengajuan Saya” yang ada di *fitur form* pengajuan. Setelah memilih *card*, maka *user* akan di arahkan ke halaman “*Form Saya*”. Di halaman *profile*, *user* dapat melihat nama lengkap, NIP, dan jabatan yang akan muncul otomatis. Selanjutnya, di halaman ini juga ada *button* “*Log Out*”. Jika *user* ingin keluar dari akun, dapat menekan *button* tersebut yang mana nantinya akan keluar pertanyaan “Apakah Anda ingin *logout*?”. Di bawah pertanyaan tersebut terdapat dua *button*, jika ingin keluar akun klik *button* hijau nya dan jika tidak ingin keluar dari akun, tekan saja *button* merah nya.

Untuk desain *user interface admin* dapat dilihat di gambar-gambar berikut.



Gambar. 9 Desain UI *Admin Halaman Login*, *Lihat Jurnal Harian*, dan *Detail Jurnal Harian*

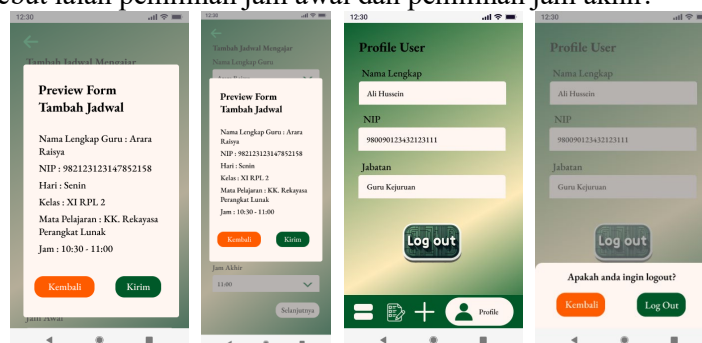
Sama seperti desain UI *user*, *admin* juga mempunyai *splash screen* dan pembagian *role*. Perbedaannya dengan *user*, *admin* harus memilih *card* “*Admin*” di halaman pembagian *role*. Nantinya akan diarahkan ke halaman *login* akun *admin*. Khusus untuk *admin*, akunnya sudah dibuat oleh *developer* di dalam *database* nya. Jadi, *admin* hanya mengisi NIP dan *password* yang diberikan untuk masuk ke dalam akunnya. Untuk bagian *admin* *fitur* pertamanya dapat melihat semua *jurnal harian* yang sudah diisi oleh semua *guru* setiap harinya. Klik saja *card* “*Lihat Jurnal Harian*”, lalu akan ada *dropdownformfield* untuk pemilihan kelas dan hari. Dapat dilihat di gambar 8, setelah *admin* memilih kelas dan hari, akan muncul otomatis *card*. *Card* tersebut akan muncul sebanyak *user* (*guru*) yang telah mengisi *jurnal* hariannya di hari tersebut. Setiap *card* akan terdapat nama *user* yang mengisi *jurnal* nya. Setelah itu, untuk melihat detail nya, *admin* dapat memilih *card* tersebut dan selanjutnya akan di arahkan ke halaman detailnya seperti gambar tersebut.



Gambar. 10 Desain UI *Admin Fitur Lihat Pengajuan* dan *Fitur Tambah Jadwal Mengajar*

Fitur selanjutnya yang dimiliki *admin* adalah dapat melihat *form* pengajuan dari semua *user* (*guru*) yang sudah mengisinya. Untuk melihatnya, klik saja *card* “*Lihat Pengajuan*”. Lalu, akan diarahkan ke halaman “*Daftar Form Pengajuan Guru*”. Setelah berada di halaman ini, nama-nama *user* akan otomatis muncul berupa

list card. Di dalam halaman daftar *form* pengajuan, *admin* dapat menggunakan fitur pencarian. Caranya hanya dengan mengetikkan nama guru yang ingin dicari, kemudian klik *icon* pencarian yang ada di dalam *form* pencarian sebelah kanan. Ini sangat bermanfaat ketika sudah banyak sekali guru yang mengajukan *form* pengajuan. Selanjutnya, untuk melihat detail pengajuannya, klik saja pada salah satu *card* yang ingin dilihat. Lalu, akan muncul halaman “Detail Form Pengajuan”. Selain itu, *admin* juga mempunyai fitur tambah jadwal mengajar *user* (guru). Untuk mengisinya klik saja *card* “Tambah Jadwal Mengajar”. Lalu, akan diarahkan ke halaman Tambah Jadwal Mengajar. Saat *admin* memilih nama guru, maka NIP guru yang bersangkutan akan langsung muncul di *form* tersebut. Selanjutnya, *admin* dapat memilih hari, kelas, dan mata pelajaran yang sesuai dengan jadwal mengajar yang ada. Di fitur inilah *admin* dapat menambahkan semua jadwal mengajar guru yang ada di Sekolah. Di sini *admin*, dipermudah dengan hanya memilih *item-item* dari tambah jadwal mengajar. Artinya *admin* tidak perlu mengetik semuanya. Setelah *admin* mengisi nama lengkap, NIP, hari, kelas, dan mata pelajaran, langkah selanjutnya adalah dua langkah terakhir untuk menyelesaikan pengisian jadwal mengajar guru. Dua langkah tersebut ialah pemilihan jam awal dan pemilihan jam akhir.



Gambar. 11 Desain UI Admin Preview Form Tambah Jadwal dan Fitur Profile

Setelah menekan *button* “Selanjutnya”, *admin* akan melihat *preview* dari *form* yang telah diisi sebelumnya. Jika dirasa sudah benar dapat menekan *button* “Kirim” dan jika dirasa masih ada kesalahan dapat menekan *button* “Kembali”. Saat *admin* menekan *button* “Kirim” (berwarna hijau) tersebut, maka selanjutnya akan melihat pemberitahuan bahwasanya telah berhasil menambahkan jadwal mengajar. Fitur terakhir dari *admin* ini ialah fitur *profile*. Di dalam halaman ini, *admin* akan melihat nama lengkap, NIP, dan jabatannya yang akan muncul secara otomatis. Lalu, jika *admin* ingin keluar dari akun, maka dapat menekan *button* “Log out”. Namun, sebelum keluar akun akan ada pertanyaan “Apakah anda ingin logout?”, jika ya klik saja *button* hijau dan jika tidak klik saja *button* merah..

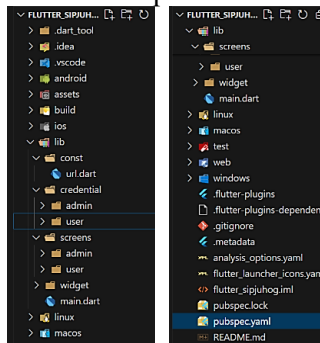
2. Pengkodean

Pada tahap ini peneliti sebagai *developer* akan membangun aplikasi *mobile* SiPJUHOG dari awal, karena *mobile* SiPJUHOG bukan aplikasi *mobile* pengembangan. Di tahap ini, peneliti memilih *framework* flutter dan bahasa dart untuk membangun aplikasi *mobile* nya. Flutter merupakan SDK untuk mengembangkan aplikasi *mobile* dengan kinerja yang tinggi untuk iOS dan Android dari satu basis kode yang dibangun oleh google [18]. Sedangkan dart adalah bahasa pemrograman untuk aplikasi pada *multiplatform* yang dikembangkan oleh Google dan dipakai untuk membangun beberapa tipe-tipe aplikasi termasuk *mobile*, *desktop*, *backend*, dan *web application* [19]. Aplikasi *mobile* ini hanya dibangun dalam bentuk android saja. Dalam tahap ini, peneliti membuat pengkodean untuk tampilan antar muka nya terlebih dahulu. Setelah itu barulah diintegrasikan ke dalam *local database* (*WampServer*) menggunakan bahasa PHP dan SQL. Perlu diketahui bahwasanya PHP (*hypertext preprocessor*) ialah salah satu dari Bahasa pemrograman yang berjalan dalam sebuah *web server* [20]. Sedangkan *structured query language* (SQL) merupakan bahasa standar untuk membangun berbagai operasi data di dalam *database* yang diantaranya mendeskripsikan tabel, memperlihatkan data dengan kualifikasi tertentu, menambahkan data tertentu, juga menghapus data [21]. Sementara *local database wampserver* ialah singkatan dari *windows*, *apache*, *mysql*, dan *php*. *WampServer* merupakan *software installer* PHP yang dengan instannya berjalan di dalam lingkungan *windows*, *php*, *apache web server*, dan *mysql* (*software* ini akan otomatis ada di *windows* kita saat proses instalasai selesai) [22].

Untuk membuat pengkodean antar muka dari *mobile* ini, dibangunnya di dalam *software* visual studio code yang mana sudah *terinstall* flutter dan dart nya (dengan semua setting di vs code juga android studio). Nama dari *project* ini adalah “flutter_sipjuhog”. Pengkodean *mobile* nya sendiri dibuat di dalam folder yang bernama “lib”. Di dalam folder tersebut terdapat *file* main.dart (*file* utama dari projek) dan empat folder lainnya. Empat folder itu adalah folder *const* (berisi *file url*), folder *credential* (berisi folder *user* dan *admin* untuk *sign up* ataupun *log in*), folder *screens* (berisi folder halaman *user* dan *admin*), dan folder *widget* (berisi *file card*). Berikut gambar dari struktur projek *mobile* “SiPJUHOG”.

1. Struktur *Project Flutter Aplikasi Mobile*

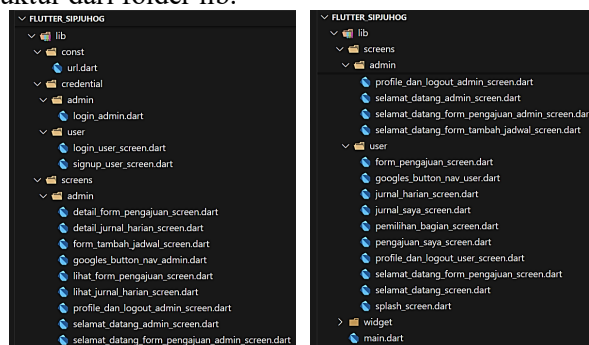
Berikut merupakan struktur proyek *mobile* menggunakan *framework* flutter dan bahasa dart. Di dalam proyek ini terdapat folder android, ios, linux, macos, web, windows, dan folder ataupun file penting lainnya. Untuk membuat *mobile*, folder yang akan sering dilihat ialah folder android saja, karena untuk membuat *permission* ataupun mengatur *gradle* dan hal lainnya akan berada di folder ini. Selain itu, dikarenakan pengerjaan seluruh codingan hanya terpusat di satu folder yang bernama lib, maka selama pengerjaan akan sering berada di folder tersebut. Sedangkan untuk keperluan penambahan gambar di dalam aplikasi *mobile*, umumnya dibuat dalam folder lain bernama “*assets*” yang mana di dalamnya mempunyai satu folder bernama “*images*” untuk menampung semua asset dari aplikasi.



Gambar. 12 Struktur *Project Flutter Aplikasi Mobile*

2. Struktur Folder “lib” (Pusat Codingan)

Di dalam folder lib ini akan dibangun semua codingan dari aplikasi *mobile*. Mulai dari fitur pertama hingga fitur terakhir, baik untuk *user* maupun *admin*. Folder *screens* dibuat untuk menampung banyaknya halaman yang akan dibangun. Di dalam folder ini terdapat dua folder lainnya yakni *user* dan *admin*. Hal tersebut bagus untuk dilakukan agar tahu yang mana saja fitur *user* dan *admin*. Lalu, nama-nama file nya lebih baik disertakan kata “*screen*” diujung nya sebelum ekstensi file dart (.dart). Contohnya jurnal_harian_screen.dart. Setiap file nya disarankan untuk membuat *class* yang *extend* ke Stateful Widget. Stateful Widget sendiri merupakan *widget* yang mempunyai status yang dapat berubah (dinamis). Maksudnya ketika ada perubahan selama menggunakan *widget* tersebut akan secara sinkron dibaca dan dapat langsung merubahnya. Pelaksana *widget* yang mempunyai tanggung jawab untuk memastikan bahwasanya *state* tersebut akan diberitahu ketika status nya berubah ialah menggunakan “*State.setState*”. Ini akan sangat membantu ketika akan membuat fitur yang terkait dengan *database*. Namun, selama pembuatan *mobile* ini rata-rata menggunakan stateful widget, karena hampir semua fitur terkait dengan *database*. Semua pengkodean antar muka dibuat menggunakan bahasa dart dengan *framework* flutter, tetapi untuk mengintegrasikan dengan *database local (wampserver)* di antar mukanya masih menggunakan bahasa dart dan *framework* flutter. Berikut merupakan struktur dari folder lib.



Gambar. 13 Struktur Folder “lib”

Selanjutnya untuk pembuatan pengkodean dalam pengintegrasian dengan *database* nya dilakukan bersamaan dengan pembangunan pengkodean antar mukanya. Itu karena fitur-fitur tersebut memiliki relasi dengan *database (read* ataupun *create)*. Contohnya saja menyimpan jurnal ataupun membaca jurnal. Pengkodeannya sendiri dibuat menggunakan bahasa PHP dan SQL. Pemilihan PHP sebagai bahasa pemrograman untuk pengintegrasian adalah karena *database* nya berjalan di *website* dan juga bahasa ini *support* untuk flutter dan dart. Sesuai dengan perkataan Laazari, dkk. [23] bahwasanya PHP mempunyai fitur intuisi, terukur, eksekusi yang efisien, *open source code*, kompatibilitas lintas *platform*, dan mendukung SQL. Artinya selaras dengan keperluan dalam pembuatan *mobile* ini bagian *database*, karena keperluan juga menggunakan bahasa SQL.

Untuk membuat pengkodean yang terkait dengan *database*, perlu membuat folder “flutter-mobile-sipjuhog” di dalam folder “www”. Di sinilah dibuat semua file yang berkaitan dengan *database* termasuk folder “*uploads*” untuk menampung foto dari jurnal harian. Setiap file diawali dengan `<?php` (di dalam sini isi dengan *query* nya, misalnya “INSERT INTO”, “SELECT”, dan lainnya) `?>`. Oleh karena *server* yang dipakai adalah *server local*, maka *server* tersebut harus dibuat menjadi *public*. Caranya dengan menggunakan “cloudflared”. Sebelum menggunakannya terlebih dahulu harus mengunduh cloudflared nya. Selanjutnya untuk menggunakannya harus membuka *command prompt* (CMD) dengan mengalihkan direktori folder yang terdapat cloudflared nya (C:\Program Files (x86)\cloudflared), kemudian ketikkan “cloudflared tunnel --url localhost:80”, tunggu sampai muncul *link* yang diberikan. Setelah mendapatkannya, inputkan *link* nya ke dalam *project mobile* di dalam folder *const* yang mempunyai file “url.dart”. Lalu, langkah selanjutnya ialah melakukan *build* aplikasi *release*. Selama *command prompt* tidak di *close*, maka pengguna dapat mengakses *database* nya.

Sedangkan untuk mendukung pembuatan pengkodean tersebut ada beberapa *package* atau *library* yang harus ditambahkan ke dalam file “pubspec.yaml”. *Package* yang dibutuhkan dalam pembuatan codingannya adalah sebagai berikut.

```
google_fonts: ^4.0.4
google_nav_bar: ^5.0.0
calendar_date_picker2: ^0.5.3
intl: ^0.18.1
image_picker: ^1.0.4
flutters_toast: ^8.0.2
http: ^0.13.6
shared_preferences: ^2.2.0
flutter_launcher_icons: ^0.13.1
rename: ^3.0.2
```

Gambar. 14 *Package* dalam Proyek Aplikasi *Mobile*

Pembuatan pengkodean dimulai dari *splash screen*, halaman pembagian *role*, halaman *sign up user*, halaman *login user*, *nav bar user*, halaman selamat datang untuk fitur jurnal harian, halaman isi jurnal harian, halaman jurnal saya untuk melihat jurnal yang telah diisi, halaman selamat datang untuk fitur *form* pengajuan, halaman ajukan untuk mengajukan *form* pengajuan, halaman pengajuan saya untuk melihat semua pengajuan yang sudah pernah diajukan, halaman *profile user*, halaman *logout user*, halaman *login admin*, halaman selamat datang fitur jurnal harian untuk *admin*, halaman selamat datang fitur *form* pengajuan untuk *admin*, halaman tambah jadwal mengajar, halaman *profile admin*, halaman *logout admin*, dan membuat *string url* untuk mengontrol *ipconfig* atau *url* dari *cloudflared*.

Tidak hanya membuat codingan antar mula *mobile* nya saja, tetapi di dalam waktu yang sama dibangun sebuah *database* memakai *software wampserver*, membuat codingan menggunakan PHP dan SQL dalam sebuah folder bernama “flutter-mobile-sipjuhog” yang menampung semua *query* untuk integrasi antar muka aplikasi *mobile* ke dalam bagian *server* nya. Codingan yang dibuat berdasarkan kebutuhan aplikasi *mobile* nya, seperti file *sign up* untuk simpan akun *user*, *login* untuk masuk akun *user*, tambah jurnal untuk *user*, *read* mata pelajaran untuk *user*, *read* jurnal untuk *user*, tambah *form* pengajuan untuk *user*, *read form* pengajuan untuk *user*, *read* data untuk *profile user*, *login* untuk masuk akun *admin*, *read all* jurnal untuk *admin*, *read all form* untuk *admin*, tambah jadwal mengajar untuk *admin*, *read* data NIP untuk *admin*, *read* data nama lengkap untuk *admin*, dan *read* data untuk *profile admin*.

Berikut merupakan hasil codingan dari aplikasi *mobile* SiPJUHOG yang sudah dibangun dan dijalankan di *handphone* serta logo aplikasi *mobile* SiPJUHOG.

1. Logo Aplikasi *Mobile* SiPJUHOG



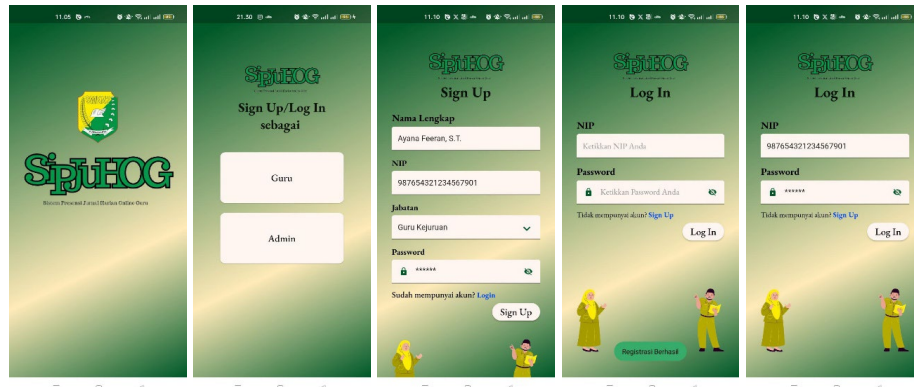
Sistem Presensi Jurnal Harian Online Guru

Gambar. 15 Logo Aplikasi *Mobile* SiPJUHOG

2. *Splash Screen*, Halaman Pembagian *Role*, Halaman *Sign up* dan Halaman *Login (User)*

Pertama saat membuka aplikasi *mobile* SiPJUHOG, *user* ataupun *admin* akan melihat *splash screen* dan pembagian *role* (*user* atau *admin*). Di dalam *splash screen* terdapat dua *item* yakni logo SMK Negeri 2 Purwakarta dan logo SiPJUHOG. Sedangkan di halaman pembagian *role* pengguna akan ditanya ingin *signup* atau *login* sebagai “Guru” atau “Admin”. Selanjutnya *user* (*guru*) memasuki halaman pendaftaran akun (*sign up*). Di sini harus mengisi nama lengkap, NIP, memilih jabatan, dan

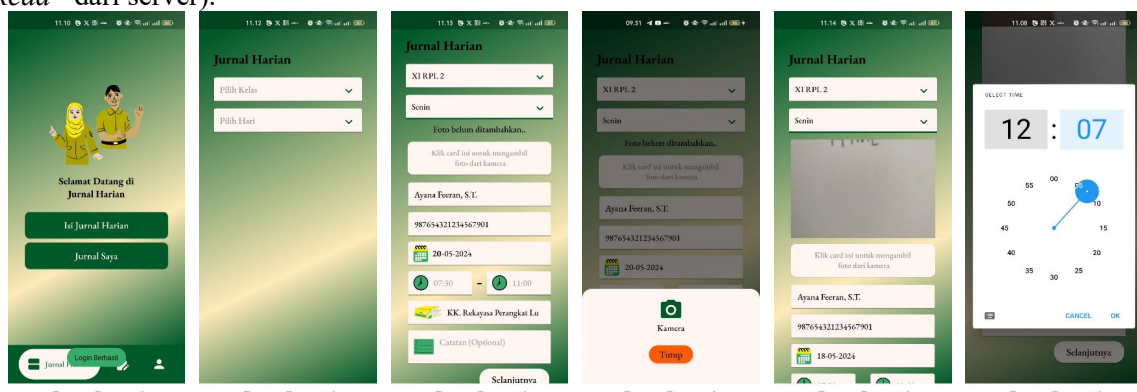
mengisi *password* seperti gambar di bawah ini. Pemilihan jabatannya sendiri ada kepala sekolah, waka kurikulum, waka kesiswaan, waka hubinmas, waka sarana dan prasaranan, guru kejuruan, dan guru mata pelajaran. Selanjutnya tekan tombol *sign up*, maka akan diarahkan ke halaman *login* dengan pemberitahuan “Registrasi berhasil” (data masuk ke *database*). Selanjutnya, pengguna harus mengisi NIP dan *password* yang telah didaftarkan sebelumnya untuk masuk ke akunnya (data yang dimasukkan haru *match* dengan data yang diinputkan). Halaman-halaman aplikasi *mobile* tersebut dapat dilihat di gambar berikut.

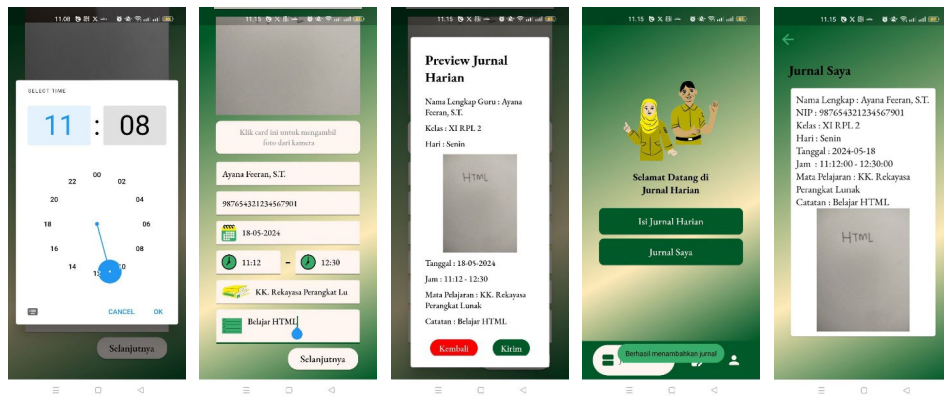


Gambar. 16 Splash Screen, Halaman Pembagian Role, Halaman Sign up dan Halaman Login (User)

3. Halaman Fitur Jurnal Harian (User)

Setelah pengguna *login*, maka yang dilihat pertama kali adalah “Selamat Datang di Jurnal Harian” dan akan melihat pemberitahuan “Login Berhasil”. Di fitur ini, pengguna dapat mengisi jurnal harian dan melihat jurnal yang sudah pernah diisi sebelumnya. Untuk mengisi jurnal harian, pengguna dapat memilih *card* “Isi Jurnal Harian”. Lalu, halaman akan berpindah ke halaman jurnal harian. Di halaman ini *user* harus memilih kelas dan hari. Jika telah mengisi kedua hal tersebut, maka akan muncul *form* lain beserta data nama lengkap, NIP, tanggal, dan mata pelajaran. Lalu, untuk pengambilan foto, klik *card* di bawah *dropdownformfiel* pemilihan hari, maka akan muncul *icon* kamera untuk mengambil foto secara langsung. Setelah itu foto akan tersimpan dilayar (jika ingin mengganti foto, klik Kembali di bagian *card* tersebut). Langkah selanjut nya yang harus dilakukan ialah memilih jam awal mengajar dan jam akhir mengajar (klik bagian *form* jam yang ditandai *icon* jam). Hal terakhir yang diisi adalah catatan (opsional). Untuk menyimpan jurnal, pengguna harus klik tombol “Selanjutnya”, lalu akan muncul *preview* jurnal yang telah diisi. *Preview* sangat bermanfaat bagi *user*, karena jika terdapat kesalahan *user* dapat memilih tombol “Kembali” untuk balik ke halaman sebelumnya dan jika tidak ada kesalahan dapat langsung memilih tombol “Kirim”. Setelah pengguna menekan tombol kirim, maka jurnal tersebut akan masuk ke dalam *database localhost* di *wampserver*. Untuk melihat semua jurnal yang telah diisi, *user* dapat memilih *card* “Jurnal Saya” di fitur jurnal harian (data jurnal di “Read” dari server).

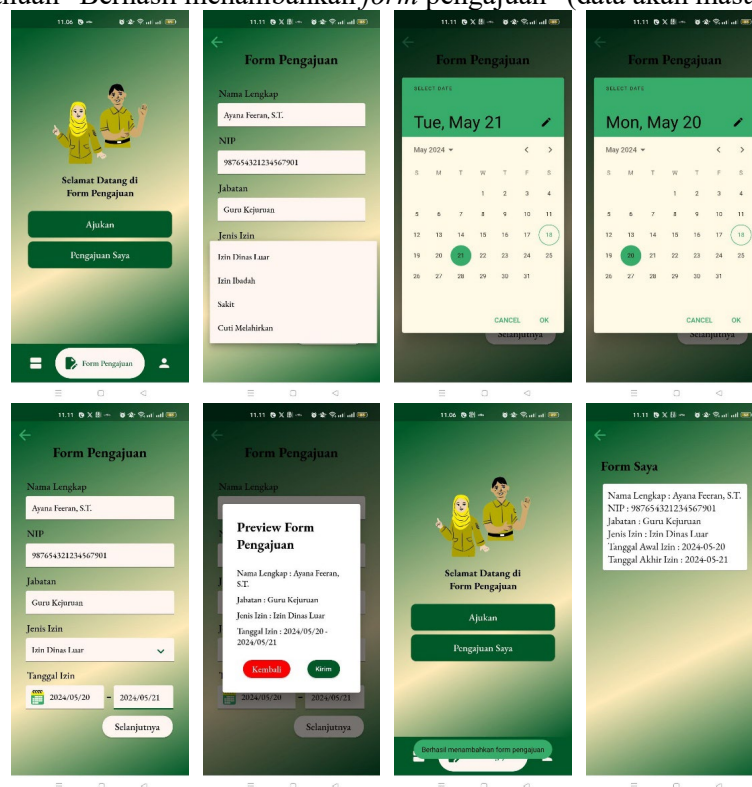




Gambar. 17 Halaman Fitur Jurnal Harian (User)

4. Halaman Fitur *Form Pengajuan* (User)

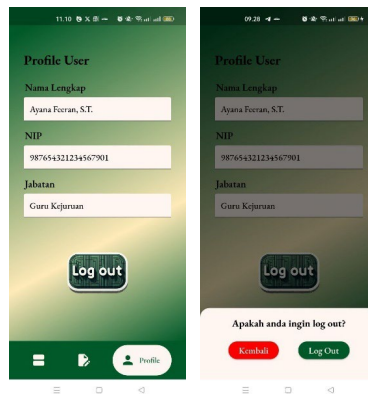
Selanjutnya setelah mengisi jurnal harian, jika *user* ingin mengajukan izin, dapat langsung beralih ke dalam fitur *form* pengajuan. Untuk mengisi *form* nya pilihlah *card* “Ajukan”, kemudian pengguna akan dialihkan ke halaman “*Form Pengajuan*”. Di sini untuk nama lengkap, NIP, dan Jabatan di “*Read*” dari data *sign up user*. Tahap selanjutnya pengguna hanya memilih jenis izin, tanggal izin, dan tanggal akhir izin. Sama seperti jurnal harian, untuk mengirimkannya klik tombol “Selanjutnya”, maka *user* akan melihat *preview*. Saat pengguna sudah merasa yakin tidak ada kesalahan, maka dapat klik tombol “Kirim”, jika belum klik tombol “Kembali”. Lalu, saat *user* berhasil mengirimkan *form* nya, maka akan muncul pemberitahuan “Berhasil menambahkan *form* pengajuan” (data akan masuk ke *wampserver*).



Gambar. 18 Fitur *Form Pengajuan* (User)

5. Halaman Fitur *Profile* dan *Log Out* (User)

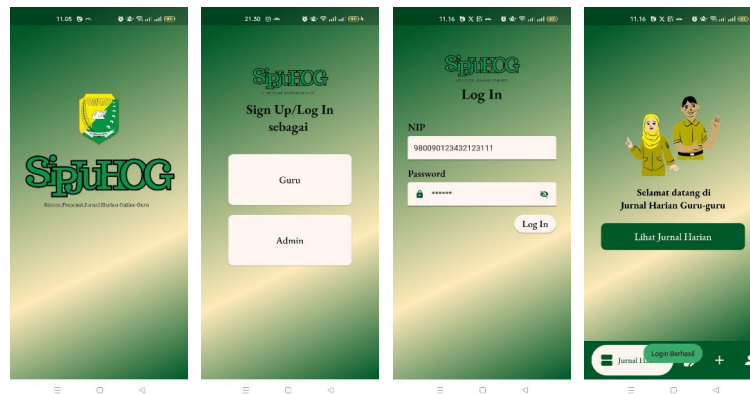
Terakhir *user* dapat melihat *profile* nya yang ada di halaman *profile*. Di halaman ini nama lengkap, NIP, dan jabatan akan langsung muncul, karena data nya di “*Read*” dari data *sign up user*. Lalu, jika pengguna ingin keluar dari akun, klik tombol “*Log out*”, kemudian akan keluar pertanyaan “Apakah anda ingin *log out*?”, jika ya klik “*Log Out*” dan jika tidak klik “*Kembali*”.



Gambar. 19 Fitur *Profile* dan *Log Out (User)*

6. Halaman Fitur *Log In (Admin)*

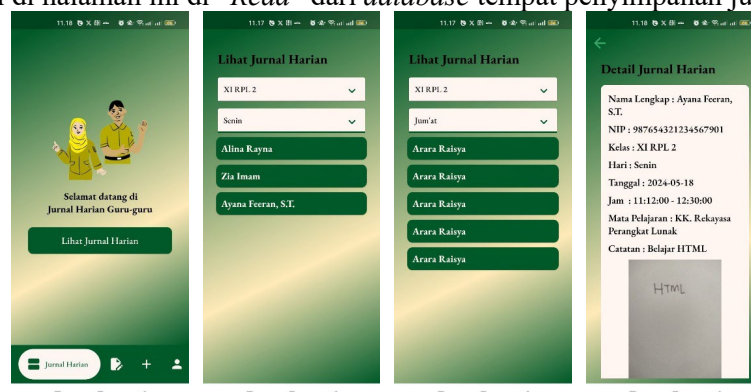
Admin mempunyai *splash screen* dan halaman pembagian *role* sama seperti *user*. Khusus untuk *admin* akan langsung ke halaman *login* (data nya sudah diinputkan oleh *developer* ke dalam *database*), jadi *admin* langsung dapat *login*. Setelah berhasil *login*, maka akan ada pemberitahuan “*Login Berhasil*”



Gambar. 20 Fitur *Log In (Admin)*

7. Halaman Fitur Jurnal Harian Bagian “*Lihat Jurnal Harian*” (*Admin*)

Setelah *login*, *admin* dapat melihat jurnal harian yang telah diisi oleh semua guru setiap harinya. Caranya adalah dengan memilih *card* “*Lihat Jurnal Harian*”, lalu akan dialihkan ke halaman tersebut. Di halaman ini, pilihlah kelas dan hari seperti yang ada di gambar berikut. Setelah itu akan muncul data nama yang sudah mengisi jurnal harian. Pilihlah salah satu *card* untuk melihat detail jurnal harian. Data yang muncul di halaman ini di “*Read*” dari *database* tempat penyimpanan jurnal harian.

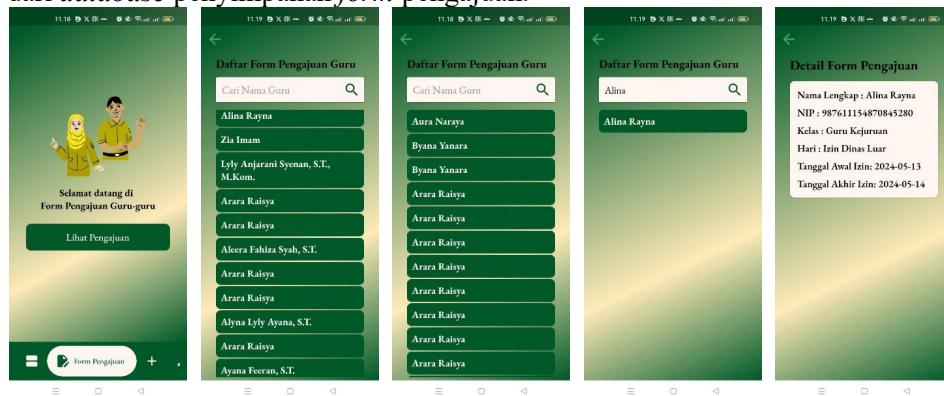


Gambar. 21 Fitur Jurnal Harian Bagian “*Lihat Jurnal Harian*” (*Admin*)

8. Fitur *Form Pengajuan* Bagian “*Lihat Pengajuan*” (*Admin*)

Selain dapat melihat jurnal harian, *admin* juga dapat melihat semua pengajuan dari guru. Untuk melihatnya *admin* dapat memilih *card* “*Lihat Pengajuan*” dan akan diarahkan ke halaman “*Daftar Form Pengajuan Guru*”. Di halaman ini data akan muncul secara otomatis. Data-data tersebut di “*Read*” dari *database* penyimpanan *form* pengajuan guru. Di sini juga dilengkapi fitur pencarian untuk mencari nama guru yang ingin dicari, jika data yang ada sudah terlalu banyak. Cara caranya hanya mengetikkan nama guru dan klik *icon* pencarian, lalu akan muncul nama sesuai yang dicari sebelumnya. Untuk

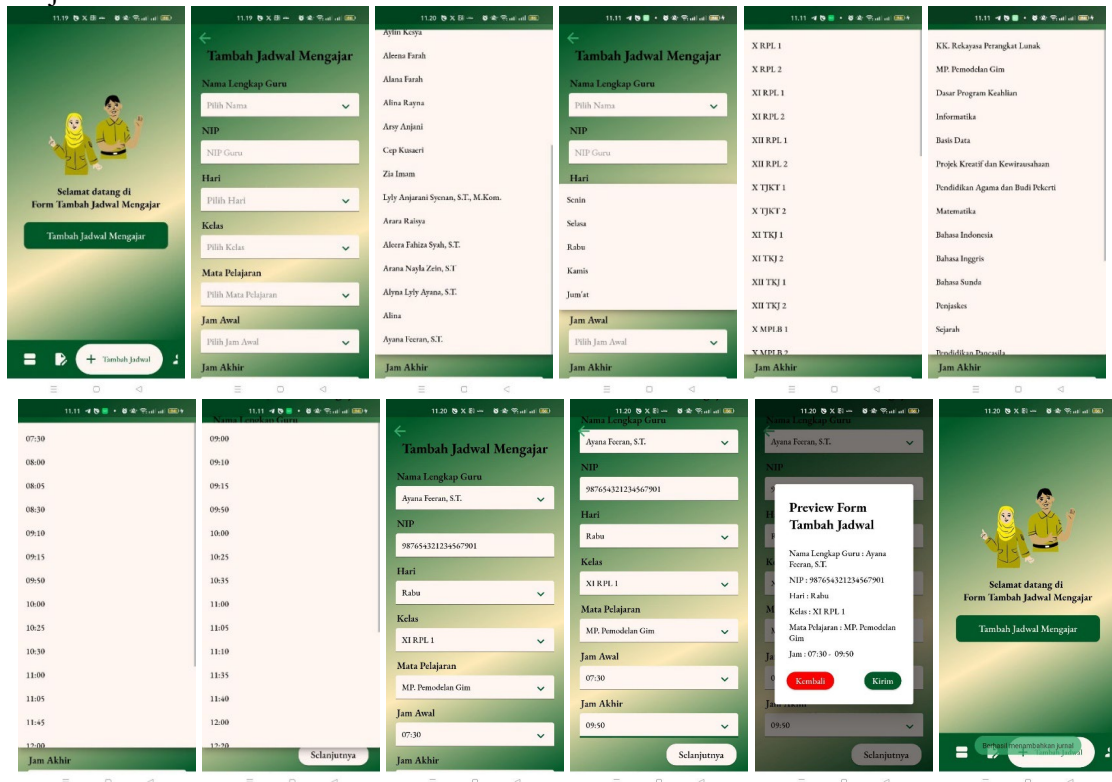
melihat detail pengajuan, klik *card* tersebut dan detail nya akan muncul. Data di dalam detail pengajuan di “*Read*” dari *database* penyimpanan *form* pengajuan.



Gambar. 22 Form Pengajuan Bagian “Lihat Pengajuan” (Admin)

9. Fitur Tambah Jadwal Mengajar (Admin)

Selain dapat melihat jurnal dan pengajuan, *admin* juga dapat menambahkan jadwal mengajar. Cara mengisi jadwalnya adalah dengan memilih *card* “Tambah Jadwal Mengajar”, lalu akan dialihkan ke halaman “Tambah Jadwal Mengajar”. Di halaman ini, *admin* akan memilih nama lengkap guru yang mana setelah terpilih nama nya, maka akan muncul NIP guru tersebut. Selain itu, *admin* juga akan memilih hari, kelas, mata pelajaran, jam awal, dan jam akhir. Setelah itu klik tombol “Selanjutnya” dan akan keluar *preview* dari jadwal mengajar guru yang bersangkutan. Jika terdapat kesalahan input salah satu *item* nya, klik saja tombol “Kembali”, jika tidak ada kesalahan klik tombol “Kirim”. Data jadwal tersebut akan masuk ke *database* yang menyimpan jadwal mengajar guru. Untuk menandai bahwasanya data sudah masuk, *admin* akan diberikan pemberitahuan bahwasanya “Berhasil menambahkan jurnal”.

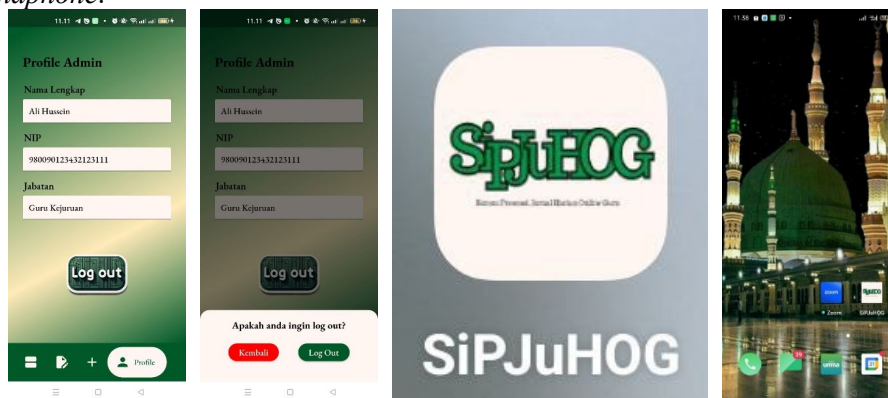


Gambar. 23 Tambah Jadwal Mengajar (Admin)

10. Fitur Profile dan Log Out (Admin)

Sama hal nya seperti *user*, *admin* juga dapat melihat *profile* nya yang ada di halaman *profile*. Di halaman ini nama lengkap, NIP, dan jabatan akan langsung muncul, karena data nya di “*Read*” dari data *log in user*. Lalu, jika pengguna ingin keluar dari akun, klik tombol “Log out”, kemudian akan keluar pertanyaan “Apakah anda ingin log out?”, jika ya klik “Log Out” dan jika tidak klik “Kembali”. Semua gambar tersebut adalah *running* dari aplikasi *mobile* SiPJUHOG di proyek “flutter_sipjuhog”. Ketika proyek selesai dibangun, maka selanjutnya adalah membangun aplikasi menjadi .apk dengan

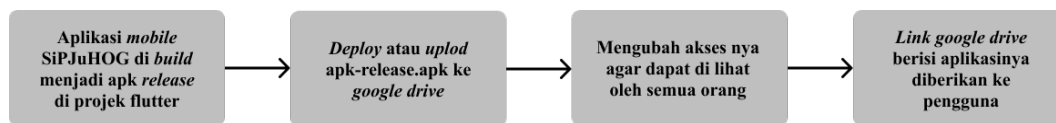
command diterminal “flutter build apk --release”. Berikut gambar dari aplikasi *mobile* yang sudah di *install* di *handphone*.



Gambar. 24 Fitur *Profile Admin* dan Aplikasi *Mobile* SiPJuHOG

E. Deployment Delivery & Feedback

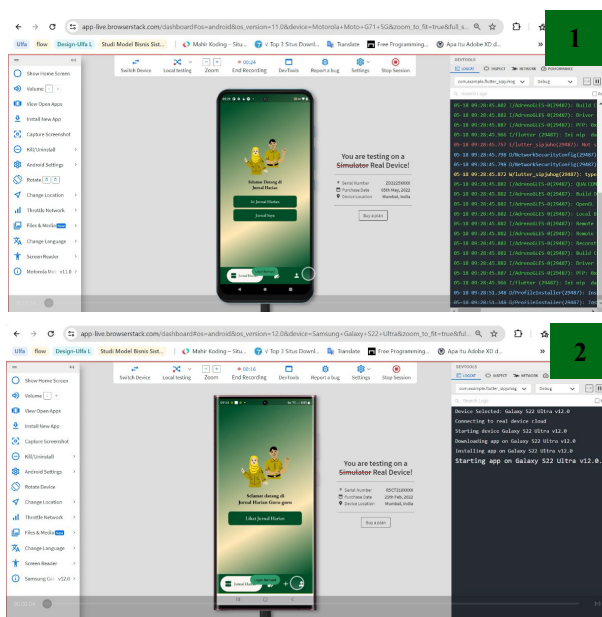
Pada tahap kelima ini dilakukan *deployment* sistem, pengujian sistem yang telah dibangun, umpan balik dari *user*, dan analisis kualitas sistem. Pengujian dilakukan untuk mengetahui apakah sistem berjalan sesuai dengan keperluan dan tujuan yang diinginkan. Setelah menyelesaikan pengkodean, maka selanjutnya aplikasi *mobile* dapat di *deploy*. Aplikasi *mobile* yang sudah dibangun akan di *build* di dalam proyek flutter nya dengan *command* “flutter build apk --release”. Setelah itu aplikasi *mobile* nya dapat di ambil dari folder flutter_sipjuhog (nama proyek aplikasi *mobile*) build\app\outputs\flutter-apk\apk-release.apk. Setelah itu, aplikasi *mobile* nya di *deploy* atau di *upload* ke *google drive* yang di *setting* dapat dilihat oleh semua orang yang punya *link* nya. Berikut alur *deployment* aplikasi *mobile* SiPJuHOG.



Gambar. 25 Alur *Deploy* Aplikasi *Mobile*

<https://drive.google.com/drive/folders/1P27kJ7RKUzKZYOOCEpJEIJNFOMKkfuut?usp=sharing> (*Link* aplikasi *mobile* SiPJuHOG).

Testing dilakukan dengan menggunakan *website* “BrowserStack” yang memiliki fitur “*App Live*” (*manual testing*). Ketika di *testing*, semua fitur di *user* dan *admin* berjalan dengan baik (tidak ada *error* UI ataupun *error* simpan data). Berikut merupakan *screenshot* saat *testing* bagian *user* (ditandai dengan nomor 1) dan *admin* (ditandai dengan nomor 2).



Gambar. 26 *Testing Manual* Aplikasi *Mobile* SiPJuHOG (*User* dan *Admin*)

Selanjutnya untuk menguji kualitas aplikasi *mobile* ini menggunakan model ISO/IEC 9126. Berikut merupakan hasil analisis kualitas dari aplikasi *mobile* SiPJuHOG.

1. Uji Fungsionalitas

Uji ini dilaksanakan untuk mendefinisikan apakah aplikasi *mobile* SiPJuHOG sudah memenuhi standar atau keperluan dari pengguna. Dalam pengujian fungsionalitas ini diperlukan instrumen untuk mengujinya. Pengujian ini melibatkan *user* secara langsung. *User* nya adalah guru-guru dari SMK Negeri 2 Purwakarta. Instrumen untuk uji fungsionalitas ini adalah seluruh fitur dari aplikasi *mobile* SiPJuHOG. Berikut merupakan instrumen dan hasil pengumpulan data yang sudah dilakukan setelah tes aplikasi *mobile* nya.

TABEL I
 INSTRUMEN UJI FUNGSIONALITAS DAN HASIL PERHITUNGAN UJI FUNGSIONALITAS

No	Fungsi	Hasil Uji	
		Berhasil	Tidak Berhasil
1	<i>Sign Up User</i>	11	0
2	<i>Login User</i>	11	0
3	<i>Login Admin</i>	11	0
4	Isi Jurnal Harian (<i>User</i>)	11	0
5	Pengambilan Foto (<i>User</i>)	11	0
6	Pemunculan Mata Pelajaran (<i>User</i>)	11	0
7	Jurnal Saya (<i>User</i>)	11	0
8	Ajukan (<i>User</i>)	11	0
9	Pengajuan Saya (<i>User</i>)	11	0
10	<i>Profile User</i>	11	0
11	<i>Log Out User</i>	11	0
12	Lihat Jurnal Harian (<i>Admin</i>)	11	0
13	Lihat Pengajuan (<i>Admin</i>)	11	0
14	Tambah Jadwal Mengajar (<i>Admin</i>)	11	0
15	<i>Profile Admin</i>	11	0
16	<i>Log Out Admin</i>	11	0
	Total	176	0

Selanjutnya untuk menghitung nilai dari fungsionalitas dapat menggunakan persamaan (1) berikut (Tohirin, dkk., 2019).

$$X = 1 - (A/B) \quad (1)$$

Perlu diketahui bahwasanya X adalah nilai uji fungsionalitas, A ialah total fungsi yang tidak berhasil (tidak sesuai), dan B merupakan total fungsi yang sedang dievaluasi. Berdasarkan perhitungan menggunakan persamaan (1) tersebut diperoleh hasil sebagai berikut.

$$X = 1 - (A/B)$$

$$X = 1 - (0/176)$$

$$X = 1 - 0$$

$$X = 1$$

Berdasarkan hasil perhitungan uji fungsionalitas tersebut dapat disimpulkan bahwasanya sistem aplikasi *mobile* SiPJuHOG mempunyai tingkatan fungsionalitas yang baik dan sesuai standar ISO/IEC 9126, karena nilai X adalah 1 ($0 \leq X \leq 1$) (Tohirin, dkk., 2019).

2. Uji Kegunaan (*Usability*)

Pengujian *usability* menggunakan kuisioner yang diberikan kepada 11 pengguna setelah *test* menggunakan aplikasi *mobile* SiPJuHOG. Jawaban dari item pertanyaan ada lima, yakni sangat tidak setuju (STS) bernilai 1, tidak setuju (TS) bernilai 2, netral (N) bernilai 3, setuju (S) bernilai 4, dan sangat setuju (SS) bernilai 5. Sebelum menghitung jumlah *usability* nya berapa, input terlebih dahulu data jawaban atas pertanyaan-pertanyaan yang sudah diberikan. Berikut merupakan rekapitulasi data jawaban dari pengguna.

TABEL II
 REKAPITULASI DATA JAWABAN PENGGUNA

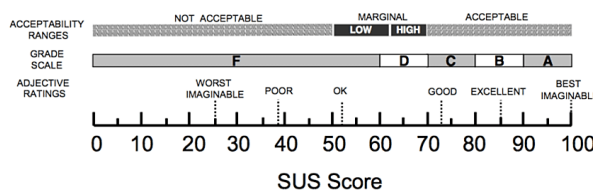
No	Item Pertanyaan	Jawaban				
		STS	TS	N	S	SS
1	Saya pikir akan sering menggunakan aplikasi <i>mobile</i> ini	0	0	0	0	11
2	Aplikasi <i>mobile</i> ini tidak terlalu rumit	0	0	0	0	11
3	Saya pikir aplikasi <i>mobile</i> ini mudah digunakan	0	0	0	0	11
4	Sepertinya saya membutuhkan bantuan untuk dapat memakai aplikasi <i>mobile</i> ini	0	0	0	11	0
5	Saya menemukan berbagai fitur dalam aplikasi <i>mobile</i> ini yang terintegrasi dengan baik	0	0	0	11	0
6	Saya menemukan ketidakkonsistenan dalam aplikasi <i>mobile</i> ini	0	9	2	0	0
7	Saya membayangkan bahawa aplikasi <i>mobile</i> ini dapat dipelajari oleh semua orang dengan cepat	0	0	0	3	8
8	Saya menemukan kerumitan dalam aplikasi <i>mobile</i> ini	10	1	0	0	0
9	Saya sangat yakin dapat menggunakan aplikasi <i>mobile</i> ini	0	0	0	1	10
10	Sepertinya saya harus belajar banyak untuk dapat menggunakan aplikasi <i>mobile</i> ini	0	0	11	0	0

Langkah selanjutnya adalah menghitung skor kontribusi yang telah diisi oleh *user*. *Item* nomor 1, 3, 5, 7, dan 9 (pertanyaan bernomor ganjil) skor kontribusinya akan dikurangi 1 dan item nomor 2, 4, 6, 8, dan 10 (pertanyaan bernomor genap) skor kontribusinya adalah 5 dikurangi posisi skala. Lalu kalikan jumlah skor kontribusi yang telah dihitung sebelumnya dengan 2.5. Berikut hasil perhitungannya.

TABEL III
 HASIL UJI USABILITY

No	Inisial Nama	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Total (Jumlah * 2,5)
1	ZIP	4	0	4	1	3	3	4	4	4	2	29	72,5
2	FN	4	0	4	1	3	3	4	4	4	2	29	72,5
3	CK	4	0	4	1	3	3	4	4	4	2	29	72,5
4	GS	4	0	4	1	3	3	4	4	4	2	29	72,5
5	M	4	0	4	1	3	3	3	3	3	2	26	65
6	RRP	4	0	4	1	3	3	4	4	4	2	29	72,5
7	ASM	4	0	4	1	3	3	4	4	4	2	29	72,5
8	IK	4	1	4	1	3	2	4	4	4	2	29	72,5
9	BHH	4	1	4	1	3	2	3	4	4	2	28	70
10	AM	4	0	4	1	3	3	4	4	4	2	29	72,5
11	AA	4	0	4	1	3	3	4	4	4	2	29	72,5
Total													787,5
Nilai SUS													71,59

Dari tabel tersebut dapat dilihat bahwasanya total yang diperoleh dari uji kegunaannya adalah sebesar 71,59. Artinya jika dilihat dari *adjective ratings* masuk kategori “GOOD”. Lalu, jika dilihat dari *grade scale* nya mendapatkan *grade C*. Sedangkan, jika dilihat dari *acceptability ranges* nya masuk kategori “ACCEPTABLE”. Seperti yang diinterpretasikan oleh gambar berikut.



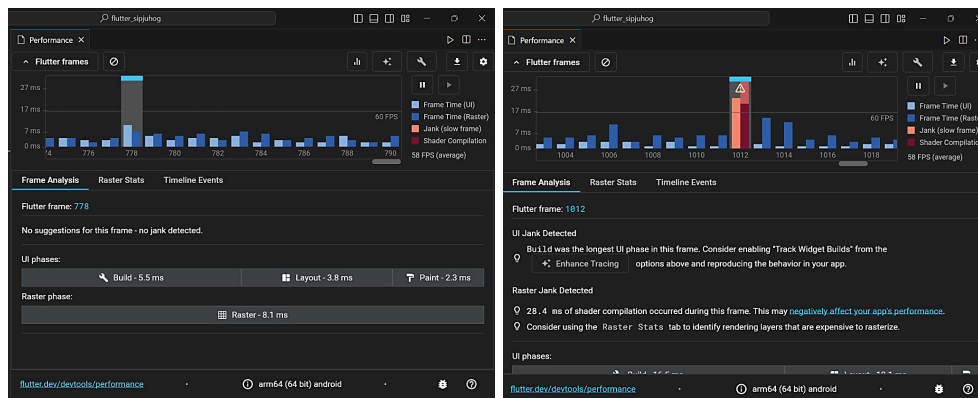
Gambar. 27 Acceptability Range SUS

3. Uji Efisiensi

Pengujian ini dilaksanakan untuk mengukur tingkat efisiensi performa dari halaman aplikasi *mobile* SiP-JuHOG. Untuk proses pengujian efisiensi nya dilakukan dengan menggunakan *tools* bawaan flutter dan dart

yaitu “DevTools”. DevTools mempunyai beberapa fitur diantaranya adalah flutter *inspector*, *timeline*, *memory*, *performance*, *debugger*, *network*, dan *logging*. Untuk melihat *performance* dari aplikasi *running* proyek nya dalam “*Mode Profile*” nantinya akan langsung muncul secara otomatis *tab performance*. Setelah itu, dapat dilihat dari gambar berikut rata-rata FPS nya adalah 58 FPS (*Frame Per Second*). FPS sendiri bekerja untuk mengukur seberapa banyak gambar atau *frame* yang diciptakan dan ditampilkan dalam satu detiknya oleh *device* (Putra, Setianingsih, dan Nugrahaeni, 2022).

Nilai 58 FPS menunjukkan bahwa aplikasi berjalan sudah cukup baik dalam hal kinerja visualnya. Walaupun seharusnya mendapatkan 60 FPS (standarnya). Berdasarkan dari nilai FPS yang didapatkan hanya 58 dapat terjadi karena performa *database* yang dimiliki, karena aplikasi *mobile* ini secara intensif mengakses *database*. Contohnya seperti mengeksekusi *query* yang banyak ataupun melakukan proses data yang besar. Artinya itu dapat menurunkan sumber daya yang ada untuk semua tugas grafisnya dan pada akhirnya akan mempengaruhi jumlah *frame per second* (FPS). Sebab lainnya karena penggunaan “*localhost*” yang mana saat terkait ke dalam *database*, maka latensi jaringan local yang tinggi menyebabkan respon aplikasi yang lambat secara keseluruhan (termasuk tampilan grafis dan FPS nya). Selain itu, penggunaan *database local* menyebabkan sumber daya CPU dan memori dapat bersaing dengan aplikasi untuk sumber daya yang sama pula dan tentunya akan mempengaruhi kinerja secara keseluruhan dari aplikasi *mobile*. Selain itu, penggunaan *mysql* juga dapat mempengaruhi FPS nya, karena sebenarnya *performance time* *postgres* lebih baik daripada *mysql* [26]. Maka dari itu untuk pembuatan *database* dalam skala yang besar dan jumlah *record* nya yang banyak akan lebih baik menggunakan *postgres*.



Gambar. 28 Performance

4. Uji Portabilitas

Pengujian portabilitas dilaksanakan dengan menjalankan atau *install* aplikasi *mobile* SiPJUHOG di sistem operasi android dengan perangkat yang berbeda. Percobaan dilakukan saat pengguna sedang mencoba aplikasi *mobile* nya. Selain itu, dicoba juga di *real device android* dalam sebuah *website* bernama “*Browserstack*”. Kriteria yang ditetapkan adalah aplikasi *mobile* SiPJUHOG dapat berjalan di *handphone* Samsung, Oppo, Vivo, Motorola. Dari hasil pengujian yang telah dilakukan dapat dilihat di tabel di bawah ini bahwasanya aplikasi *mobile* SiPJUHOG dapat berjalan di semua *handphone* tersebut.

TABEL IV
 HASIL UJI PORTABILITAS

Perangkat	Hasil Uji	
	Berhasil	Gagal
Samsung Galaxy S22 Ultra	✓	
Oppo Reno 4	✓	
Vivo V21	✓	
Moto G71 5G (Motorola)	✓	
Moto G9 Play (Motorola)	✓	

Saat melakukan *testing* kepada pengguna, pengguna memberikan jабaran responnya kepada *developer*. Umpan balik ini disampaikan setelah pengguna menyelesaikan *testing* nya. Pengguna memberikan respon mengenai pengaturan *navigator* halaman setelah menekan tombol “*Log Out*” yakni lebih baiknya menuju ke halaman pembagian *role* (halaman pengguna memilih sebagai *user* atau *admin*). Selain itu, respon lainnya adalah dapat *sign up* dengan nama lebih dari dua suku kata dan dapat menambahkan gelar juga diujung nama lengkapnya. Hasil dari umpan balik pengguna sudah *developer* perbaiki dengan sebagaimana mestinya.

Berdasarkan hasil dan pembahasan dari pembuatan aplikasi *mobile* SiPJUHOG terlihat bahwasanya *mobile* ini mempunyai beberapa fitur yakni, fitur *sign up*, fitur *login*, fitur jurnal harian berisi fitur kamera untuk bukti presensi jurnal harian yang langsung terdapat keterangan jadwal mengajar (nama, NIP, mata pelajaran), tinggal menambahkan jam awal dan jam akhir, serta terdapat catatan yang dapat diisi oleh guru (opsional). Selain itu, untuk *form* pengajuan sendiri akan memudahkan guru dalam pengajuan cuti dan pengajuan izin lainnya. Di dalam halaman *form user* hanya tinggal memilih jenis izin, memasukkan tanggal awal izin, dan tanggal akhir izin. Selain itu, *user* dapat melihat semua jurnal yang telah diisi di “Jurnal Saya”, semua *form* yang telah diisi di “Pengajuan Saya”, *profile*, dan *log out* nya. Fitur-fitur tersebut akan ada dibagian sisi *user*. Sedangkan di bagian admin terdapat fitur *login*, fitur melihat presensi jurnal harian guru, fitur melihat *form* pengajuan, fitur tambah jadwal mengajar, *profile*, dan *log out*. Seluruh fitur tersebut dibangun untuk menyelesaikan beberapa masalah yang ada, yakni adanya kesenjangan sosial antara ASN dan non-ASN perihal presensi yang ada di Sekolah. masalah presensi guru di Kelas yang ditulis secara langsung pada agenda kelas (menjadi permasalahan bagi pihak sekolah, karena pihak sekolah tidak dapat memantau secara langsung kehadiran guru-guru tersebut di kelas), dan belum terdapat sistem presensi jurnal harian guru di SMK Negeri 2 Purwakarta, sehingga kehadiran guru di kelas belum dapat terpantau langsung oleh pihak sekolah. Jika dibandingkan dengan hasil penelitian dari peneliti sebelumnya, tentulah sangat berbeda, karena dalam penelitian tersebut dijelaskan bahwasanya pengelolaan buku administrasi guru masih manual yang menyebabkan sering hilang data, membutuhkan waktu yang lam dalam mencatat administrasi guru, tidak mencatat dengan alasan karena lupa, buku ketinggalan, dan rusak ataupun hilang [1]. Lalu, fitur yang dibuat juga sangat berbeda yakni, fitur bagian *user* nya ialah *form login*, *form* pendaftaran, *form* utama, *form* agenda (pengisian agenda), *form* lihat jadwal, *form* lihat agenda, *form* tentang, *form* logout. Sedangkan aplikasi admin memiliki fitur *form login*, *form* menu utama, *form* data *user*, *form* tambah *user*, *form* jadwal, *form* tambah jadwal, dan *form* agenda.

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan yang ada dan telah dijabarkan sebelumnya, maka dapat disimpulkan bahwasanya : a) Aplikasi *Mobile* SiPJUHOG dibangun menggunakan bahasa dart dan *framework* flutter. Selain itu menggunakan bahasa PHP dan SQL dengan *server* nya menggunakan *wampserver*. Berdasarkan hasil pengujian manual (manual testing) menggunakan “Browserstack” mendapatkan hasil yang baik, karena semua fungsi yang ada di aplikasi *mobile* SiPJUHOG dapat berjalan dengan baik dan lancar. b) Hasil pengujian (*testing*) aplikasi *mobile* secara langsung kepada *user* (uji fungsionalitas) memberikan hasil yang baik yang mana berhasil melalui semua instrumen fungsi dari *mobile* ini (tidak ada yang gagal). Selanjutnya hasil uji *usability* nya mendapat nilai total 71,59 yang mana memperoleh kategori “GOOD” (*adjective ratings*), kemudian memperoleh *grade C* (*grade scale*), dan terakhir memperoleh kategori “ACCEPTABLE” (*acceptability ranges*). Lalu, untuk uji efisiensi menggunakan “DevTools” nya mendapatkan hasil rata-rata FPS (*frame per second*) sebesar 58 yang dapat diartikan bahwasanya aplikasi berjalan sudah cukup baik dalam hal kinerja visualnya. Terakhir dari hasil uji portabilitas nya yang dilakukan ketika *user* mencoba aplikasi *mobile* dan menggunakan “BrowserStack” mendapatkan hasil yang sesuai kriteria, yakni berjalan di *handphone* Samsung, Oppo, Vivo, Motorola.

UCAPAN TERIMA KASIH

Terima kasih kepada semua pihak yang telah membimbing dan mengarahkan dalam penelitian artikel ini.

DAFTAR PUSTAKA

- [1] I. Muslimin and H. Sopian, “MEMBANGUN APLIKASI AGENDA HARIAN GURU BERBASIS ANDROID (STUDI KASUS DI SMP BHAKTI MULYA BANJARAN),” *J. Inform.*, vol. 6, no. 1, pp. 59–73, 2019, [Online]. Available: <https://unibba.ac.id/ejournal/index.php/computing/article/view/32%0Ahttp://ejournal.unibba.ac.id/index.php/computing/article/view/32/30>.
- [2] F. F. Salsabila and V. F. Salsabilla, “Analisis Faktor-faktor yang Mempengaruhi Kinerja Administratif Guru dan Staf Pendukung di Sekolah (SMP Negeri 28 Surabaya),” *Pubmedia J. Penelit. Tindakan Kelas Indones.*, vol. 1, no. 3, p. 14, 2024, doi: 10.47134/ptk.v1i3.417.
- [3] R. F. Pradipta, H. I. Oktaviani, D. A. Dewantoro, and Riza, “Peningkatan Mutu Pengelolaan Data Administrasi Pembelajaran melalui Sistem Administrasi Sekolah Inklusi (SIADSI) Berbasis Website bagi Guru Pendamping Khusus,” *JURPIKAT (Jurnal Pengabd. Kpd. Masyarakat)*, vol. 3, no. 3, pp. 505–512, 2022, doi: 10.37339/jurpikat.v3i3.1033.
- [4] Syakbaniansyah, Norjanah, and Syahrani, “Penyusunan Administrasi Guru,” *J. Al-Risalah*, vol. 17, no. 1, pp. 44–53, 2021, [Online]. Available: <https://jurnal.stairakha-amuntai.ac.id/index.php/alris/article/view/207>.
- [5] D. Y. S. Mustafa Kamal Rokan, “Analisis Efektivitas Penggunaan QRIS (Quick Response-Code Indonesian Standard) Untuk Mendukung Paperless Di PT. Bank Syariah Indonesia KCP Medan Padang Bulan,” *J. Econ. Curr. Study*, vol. 4, no. 2, pp. 1–11, 2022, doi: 10.51178/jecs.v4i2.664.
- [6] S. Maesaroh, B. Bahagia, and K. Kamalludin, “Strategi Menumbuhkan Literasi Lingkungan Pada Siswa,” *J. Basicedu*, vol. 5, no. 4, pp. 1998–2007, 2021, [Online]. Available: <https://jbasic.org/index.php/basicedu/article/view/1048>.
- [7] Y. Megasyah and H. Hardian, “Analisis dan Perancangan Sistem Informasi Perpustakaan Dengan Teknologi Java dan MySQL Studi Kasus SMA SANTA MARIA 3 CIMAHI,” *Naratif (Jurnal Ilm. Nas. Ris. Apl. dan Tek. Inform.)*, vol. 1, no. 02, pp. 47–53, 2019.

- [8] S. F. Nasrulloh and M. Agustina, "Rancangan Bangun Aplikasi Catatan Pelaksanaan Pembelajaran Harian (CPPH) untuk Guru berbasis Web," vol. 5, no. 2, pp. 329–338, 2021.
- [9] R. Anjeli and Y. Budi, "APLIKASI JURNAL MENGAJAR GURU BERBASIS WEB (Studi Kasus SD Negeri 013 Rambah Samo)," *RJOCS (Riau J. Comput. Sci.*, vol. 9, no. 1, pp. 20–27, 2023.
- [10] H. M. Junaidi and A. Ghofur, "RANCANG BANGUN APLIKASI E-HARIAN GURU PADA SMKN 1 SUNGAI PINANG BERBASIS WEB," *Pranala*, vol. 17, no. 1, pp. 13–18, 2022.
- [11] I. Setiawan, R. Waluyo, and W. Suseno, "Perancangan Aplikasi Agenda Kegiatan Guru Berbasis Website," *Infotekmesin*, vol. 13, no. 1, pp. 1–7, 2022, doi: 10.35970/infotekmesin.v13i1.576.
- [12] K. Islah, "PELUANG DAN TANTANGAN PEMANFAATAN TEKNOLOGI BIG DATA UNTUK MENINGTEGRASIKAN PELAYANAN PUBLIK PEMERINTAH," *J. Reformasi Adm.*, vol. 5, no. 1, pp. 130–138, 2018, [Online]. Available: <http://ojs.stiami.ac.id/index.php/reformasi/article/view/272%0Ahttp://ojs.stiami.ac.id/index.php/reformasi/article/viewFile/272/162>.
- [13] K. C. Budiayatno, "TRANSFORMASI DIGITAL SEBAGAI BAGIAN DARI STRATEGI PEMASARAN DI RUMAH SAKIT SILOAM PALANGKA RAYA TAHUN 2020," *J. Adm. Rumah Sakit Indones.*, vol. 8, no. 2, pp. 66–73, 2022, doi: 10.7454/arsi.v8i2.5547.
- [14] O. Marshella Febriani, A. Setya Putra, and B. Karina, "Implementasi Sistem Distribusi Pupuk di PT. Gresik Cipta Sejahtera Lampung Berbasis Mobile," *J. Tek.*, vol. 15, no. 2, pp. 273–279, 2021.
- [15] V. Sahfitri, "Prototype E-Katalog Dan Peminjaman Buku Perpustakaan Berbasis Mobile," *J. SISFOKOM*, vol. 08, no. 02, pp. 165–171, 2019.
- [16] E. Muchtar and Effiyaldi, "ANALISIS DAN PERANCANGAN SISTEM INFORMASI MANAJEMEN SURAT MASUK DAN SURAT KELUAR PADA DINAS KEPENDUDUKAN DAN," *J. Manaj. Sist. Inf.*, vol. 4, no. 2, pp. 193–204, 2019.
- [17] S. Tazkiyah and A. Arifin, "PERANCANGAN UI/UX PADA WEBSITE LABORATORIUM ENERGY MENGGUNAKAN APLIKASI FIGMA," *J. Teknol. Terpadu*, vol. 8, no. 2, pp. 72–88, 2022, [Online]. Available: <https://media.neliti.com/media/publications/493730-water-ph-and-turbidity-control-system-in-0a553e14.pdf>.
- [18] S. Tjandra and G. S. Chandra, "Pemanfaatan Flutter dan Electron Framework pada Aplikasi Inventori dan Pengaturan Pengiriman Barang," *J. Inf. Syst. Hosp. Technol.*, vol. 2, no. 02, pp. 76–81, 2020, doi: 10.37823/insight.v2i02.109.
- [19] S. Sinha, *Quick Start Guide to Dart Programming*. Howrah, West Bengal, India: Apress, 2020.
- [20] A. Mubarak, "Rancang Bangun Aplikasi Web Sekolah Menggunakan Uml (Unified Modeling Language) Dan Bahasa Pemrograman Php (Php Hypertext Preprocessor) Berorientasi Objek," *JIKO (Jurnal Inform. dan Komputer)*, vol. 2, no. 1, pp. 19–25, 2019, doi: 10.33387/jiko.v2i1.1052.
- [21] A. A. Aqham, *Managemen Sistem Basis Data (SQL dan MySql)*. Semarang: Yayasan Prima Agus Teknik Redaksi, 2021.
- [22] R. D. Mandasari, "Aplikasi WAP dengan WML, PHP & MySQL dalam Sistem Informasi Akademik," *Comput. Sci.*, vol. 1, no. 1, pp. 35–43, 2021, doi: 10.31294/coscience.v1i1.179.
- [23] M. Laaziri, K. Benmoussa, S. Khouliji, and M. L. Kerkeb, "A Comparative study of PHP frameworks performance," *Procedia Manuf.*, vol. 32, pp. 864–871, 2019, doi: 10.1016/j.promfg.2019.02.295.
- [24] T. Tohirin, W. Al Mauludyansah, S. E. Setyawan, and S. R. Widiyanto, "Analisis Kualitas dan Penerapan Software Quality Assurance pada Situs web e-Clinic Menggunakan Model ISO/IEC 9126," *J. MULTINETICS*, vol. 5, no. 2, pp. 52–58, 2019, doi: 10.32722/multinetics.v5i2.2761.
- [25] S. E. Putra, C. Setianingsih, and R. A. Nugrahaeni, "Deteksi Pelanggaran Parkir Pada Bahu Jalan Tol Dengan Intelligent Transportation System Menggunakan Algoritma Ssd Parking Violation Detection On The Roadside Of Toll Roads With The Intelligent Transportation System Using Ssd Algorithm 1st," *e-Proceeding Eng.*, vol. 9, no. 3, pp. 1059–1063, 2022.
- [26] A. D. Praba and M. Safitri, "STUDI PERBANDINGAN PERFORMANSI ANTARA MYSQL DAN POSTGRESQL," *J. Khatulistiwa Inform.*, vol. 8, no. 2, pp. 88–93, 2020, doi: 10.31294/jki.v8i2.8851.